Quantum CISC Compilation by Optimal Control and Scalable Assembly of Complex Instruction Sets beyond Two-Qubit Gates

T. Schulte-Herbrüggen,* A. Spörl, and S.J. Glaser
Department of Chemistry, Technical University Munich, D-85747 Garching, Germany

(Dated: 22nd December 2008)

We present a quantum CISC compiler and show how to assemble complex instruction sets in a scalable way. Enlarging the toolbox of universal gates by optimised complex multi-qubit instruction sets thus paves the way to fight relaxation for realistic experimental settings.

Compiling a quantum module into the machine code for steering a concrete quantum hardware device lends itself to be tackled by means of optimal quantum control. To this end, there are two opposite approaches: (i) one may use a decomposition into the restricted instruction set (RISC) of universal one- and two-qubit gates and translate them into the machine code, or (ii) one may prefer to generate the entire target module as a complex instruction set (CISC) directly by evoltution under drift and available controls. Here we advocate direct compilation up to the limit of system size a classical high-performance parallel computer cluster can reasonably handle. For going beyond these limits, i.e. for large systems, we propose a combined way, namely (iii) to make recursive use of medium-sized building blocks generated by optimal control in the sense of a quantum CISC compiler.

The advantage of the method over standard RISC compilations into one- and two-qubit universal gates is explored on the parallel cluster HLRB-II (with a total LINPACK performance of 63.3 TFlops/s) for the quantum Fourier transform, the indirect SWAP gate as well as for multiply-controlled NOT gates. Implications for upper limits to time complexities are also derived.

PACS numbers: 03.67.-a, 03.67.Lx, 03.65.Yz, 03.67.Pp; 82.56.-b, 82.56.Jn, 82.56.Dj, 82.56.Fk

Introduction

Richard Feynman's seminal conjecture of using experimentally controllable quantum systems to perform computational tasks [1, 2] roots in reducing the complexity of the problem when moving from a classical setting to a quantum setting. The most prominent pioneering example being Shor's quantum algorithm of prime factorisation [3, 4] which is of polynomial complexity (BQP) on quantum devices instead of showing non-polynomial complexity on classical ones [5]. It is an example of a class of quantum algorithms [6, 7] that solve *hidden subgroup problems* in an efficient way [8], where in the Abelian case, the speed-up hinges on the quantum Fourier transform (QFT). Whereas the network complexity of the fast Fourier transform (FFT) for n classical bits is of order $O(n2^n)$ [9, 10], the QFT for n qubits shows a complexity of order $O(n^2)$. Moreover, Feynman's second observation that quantum systems may be used to efficiently predict the behaviour of other quantum systems has inaugurated a branch of research dedicated to Hamiltonian simulation [11, 12, 13, 14, 15, 16].

For implementing a quantum algorithm in an experimental setup, local operations and universal two-qubit quantum gates are required as a minimal set ensuring every unitary module can be realised [17]. More recently, it turned out that generic qubit and qudit pair interaction Hamiltonians suffice to complement local actions to universal controls [18, 19]. Common sets of quantum computational instructions comprise (i) local operations such as the Hadamard gate, the phase gate and (ii) the entangling operations CNOT, controlled-phase gates, $\sqrt{\text{SWAP}}$, $i\,\text{SWAP}$ as well as (iii) the SWAP operation. The number of elementary gates required for implementing a quantum module then gives the network or gate complexity.

As is well known, a generic n-qubit generic operation requires exponentially many two-qubit gates to be implemented exactly [20, 21], the complexity being $O(n4^n)$. Yet, as has been pointed out by Barenco *et al.*, many quantum computationally pertinent gates can be decomposed into a number of one- and two-qubit gates *increasing linearly* with the number of qubits. At the expense of a single ancilla qubit this also holds for multiply controlled unitary gates [20] tantamount to error correction. For an overview, see e.g. [22, 23, 24]. Moreover, Blais [25] showed how to implement the QFT with linear gate complexity. Later, Solovay ([26] quoted in [27] and [22]) and then Kitaev addressed the problem to approximate arbitrary unitary gates by polynomially long 2-qubit gate sequences up to a given precision [23, 27]. More recently the bounds of approximating an arbitrary unitary were taken down to a polynomial of sixth-order in the number of qubits and of third order in the geodesic distance of the unitray to unity [28]. Differential geometric aspects in terms of Finsler metrics have been raised in [29].

However, gate complexity often translates into too coarse an estimate for the actual time required to implement a quantum module (see e.g. [30, 31, 32]), in particular, if the time scales of a specific experimental setting have to be matched. Instead, effort has been taken to give upper bounds on the actual time complexity [33], e.g., by way of numerical optimal control [34].

*Electronic address: tosh@ch.tum.de

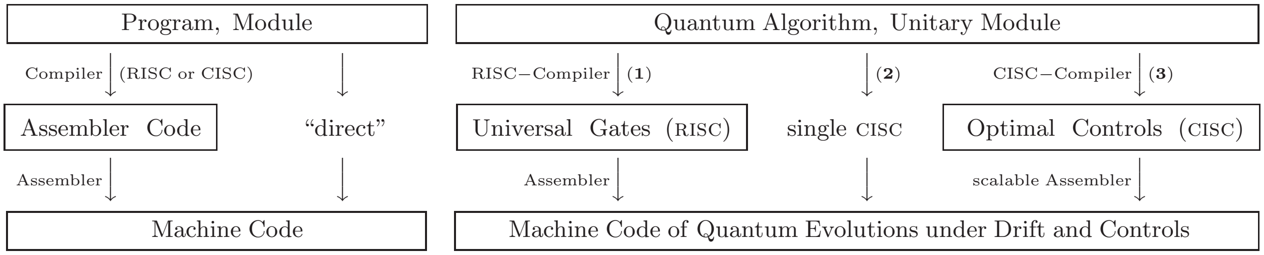

Figure 1: Compilation in classical computation (left) and quantum computation (right). Quantum machine code has to be time-optimal or protected against relaxation, otherwise the coherent superpositions are wiped out. A quantum RISC-compiler (1) by universal gates leads to unnecessarily long machine code. Direct CISC-compilation into a single pulse sequence (2) exploits quantum control for a near time-optimal quantum machine code. Its classical complexity is NP, so direct compilation by numerical optimal control resorting to a classical computer is unfeasible for large quantum systems. The third way (3) promoted here pushes quantum CISC-compilation to the limits of classical supercomputer clusters and then assembles the multi-qubit complex instructions sets recursively into time-optimised or relaxation-protected quantum machine code.

Interestingly, in terms of quantum control theory, the *existence of universal gates* is equivalent to the statement that the quantum system is *fully controllable* as has first been pointed out in Ref. [35]. This is, e.g., the case in systems of n spin-$\frac{1}{2}$ qubits that form Ising-type weak-coupling topologies described by arbitrary connected graphs [36, 37]. Therefore the usual approach to quantum compilation in terms of local plus universal two-qubit operations [38, 39, 40, 41, 42] lends itself to be complemented by optimal-control based direct compilation into machine code: it may be seen as a technology-dependent optimiser in the sense of Ref. [41], however, tailored to deal with more complex instruction sets than the usual local plus two-qubit building blocks. Not only is it adapted to the specific experimental setting, it also allows for fighting relaxation by either being near timeoptimal or by exploiting relaxation-protected subspaces [43]. Devising quantum compilation methods for optimised realisations of given quantum algorithms by admissible controls is therefore an issue of considerable practical interest. Here it is the goal to show how quantum compilation can favourably be accomplished by optimal control: the building blocks for gate synthesis will be extended from the usual set of restricted local plus universal two-qubit gates to a larger toolbox of *scalable* multi-qubit gates tailored to yield high fidelity in short time given concrete experimental settings.

Quantum Compilation as an Optimal Control Task

As shown in Fig. 1, the quantum compilation task can be addressed following different principle guidelines: **(1)** by the standard decomposition into local operations and universal two-qubit gates, which by analogy to classical computation was termed *reduced instruction set* quantum computation (RISC) [44] or **(2)** by using direct compilation into one single *complex instruction set* (CISC) [44]. The existence of a such a single effective gate is guaranteed simply by the unitaries forming a group: a sequence of local plus universal gates is a product of unitaries and thus a single unitary itself.

As a consequence, CISC quantum compilation lends itself for resorting to numerical optimal control (on clusters of classical computers) for translating the unitary target module directly into the 'machine code' of evolutions of the quantum system under combinations of the drift Hamiltonian H_0 and experimentally available controls H_j.

In a number of studies on quantum systems up to 10 qubits, we have shown that direct compilation by gradient-assisted optimal control [34, 45, 46] allows for substantial speed-ups, e.g., by a factor of 5 for a CNOT and a factor of 13 for a Toffoli-gate on coupled Josephson qubits [46]. However, the direct approach naturally faces the limits of computing quantum systems on classical devices: upon parallelising our C++ code for high-performance clusters [47], we found that extending the quantum system by one qubit increases the CPU-time required for direct compilation into the quantum machine code of controls by roughly a factor of eight. So the classical complexity for optimal-control based quantum compilation is NP.

Therefore, here we advocate a third approach **(3)** that uses direct compilation into multi-qubit complex instruction sets up to the CPU-time limits of optimal quantum control on classical computers: these building blocks are designed such as to allow for recursive scalable quantum compilation in large quantum systems (i.e. those beyond classical computability). In particular, the complex instruction sets may be optimised such as to fight relaxation by being near time-optimal, or, moreover, they may be devised such as to fight the specific imperfections of an experimental setting.

Controllability

Before turning to optimal-control based CISC quantum compilation in more detail, it is important to ensure the quantum control system characterised by $\{H_0\} \cup \{H_j\}$ is in fact *fully controllable.*

Hamiltonian quantum dynamics following Schrödinger's equation for the unitary image of a complete basis set of 'state vectors' representing a quantum gate

$$|\dot{\psi}(t)\rangle \;=\; -i\big(H_d + \sum_{j=1}^{m} u_j(t) H_j\big)\;|\psi(t)\rangle \qquad (1)$$

$$\dot{U}(t) \;=\; -i\big(H_d + \sum_{j=1}^{m} u_j(t) H_j\big)\;U(t) \quad , \qquad (2)$$

resembles the setting of a standard *bilinear control system* with state $X(t)$, drift A, controls B_j, and control amplitudes $u_j \in \mathbb{R}$ reading

$$\dot{X}(t) = \big(A + \sum_{j=1}^{m} u_j(t) B_j\big)\;X(t) \quad , \qquad (3)$$

where $X(t) \in GL_N(\mathbb{C}\,)$ and $A, B_j \in \text{Mat}\,_N(\mathbb{C}\,)$. Clearly in the dynamics of closed quantum systems, the system Hamiltonian H_d is the drift term, whereas the H_j are the control Hamiltonians with $u_j(t)$ as control amplitudes. In systems of n qubits, $|\psi\rangle \in \mathbb{C}^{\,2^n}$, $U \in SU(2^n)$, and $i\,H_\nu \in \mathfrak{su}(2^n)$.

A system is *fully operator controllable*, if to every initial state ρ_0 the entire unitary orbit $\mathcal{O}_{\rm U}(\rho_0) := \{U\rho_0 U^\dagger \,|\, U \in SU(N)\}$ can be reached. With density operators being Hermitian this means any final state $\rho(t)$ can be reached from any initial state ρ_0 as long as both of them share the same spectrum of eigenvalues.

As established in [48], the bilinear system of Eqn. 2 is fully controllable if and only if the drift and controls are a generating set of $\mathfrak{su}(N)$ by way of the commutator, i.e., $\langle H_d, H_j \,|\, j = 1, 2, \ldots, m\rangle_{\rm Lie} = \mathfrak{su}(N)$.

Example 1 Consider a system of n weakly coupled spin-$\frac{1}{2}$ qubits. Let $\sigma_x = \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$, $\sigma_y = \left(\begin{smallmatrix} 0 & -i \\ i & 0 \end{smallmatrix}\right)$, $\sigma_z = \left(\begin{smallmatrix} 1 & 0 \\ 0 & -1 \end{smallmatrix}\right)$ be the Pauli matrices. In n spins-$\frac{1}{2}$, a σ_{kx} for spin k is tacitly embedded as $\mathbb{1} \otimes \cdots \mathbb{1} \otimes \sigma_x \otimes \mathbb{1} \otimes \cdots \mathbb{1}$ where σ_x is at position k. The same holds for σ_{ky}, σ_{kz}, and in the weak coupling terms $\sigma_{kz}\sigma_{\ell z}$ with $1 \leq k < \ell \leq n$.

Now a system of n qubits is fully controllable [36], if e.g. the control Hamiltonians H_j comprise the Pauli matrices $\{\sigma_{kx}, \sigma_{ky} \,|\, k = 1, 2, \ldots n\}$ on every single qubit selectively and the drift Hamiltonian H_d encompasses the Ising pair interactions $\{J_{k\ell}\;(\sigma_{kz}\sigma_{\ell z})/2 \,|\, k < \ell = 2, \ldots n\}$, where the coupling topology of $J_{k\ell} \neq 0$ may take the form of any connected graph. This theorem has meanwhile been generalised to other coupling types [49, 50].

In view of the compilation task in quantum computation we get the following synopsis:

Corollary 1 *The following are equivalent:*

(1) in a quantum system of n coupled spins-$\frac{1}{2}$, the drift H_d and the controls H_j form a generating set of $\mathfrak{su}(2^n)$;

(2) the quantum system is operator controllable (in the sense of Ref. [51]);

(3) every unitary transformation $U \in SU(2^n)$ can be realised by that system;

(4) there is a set of universal quantum gates for the quantum system.

Proof: The equivalence of (1) and (2) relies on the unitary group being a compact connected Lie group: compact connected Lie groups have no closed *subsemigroups* that are no groups themselves [48]. Moreover, in compact connected Lie groups the exponential mapping is surjective, hence (1) ⇒ (3). Assertions (3) and (4) just re-express the same fact in different terminology. ■

Scope and Organisation of the Paper

The purpose of this paper is to show that optimal control theory can be put to good use for devising multi-qubit building blocks designed for scalable quantum computing in realistic settings. Note these building blocks are no longer meant to be universal *in the practical sense* that any arbitrary quantum module should be built from them (plus local controls). Rather they provide specialised sets of complex instructions tailored for breaking down typical tasks in quantum computation with substantial speed gains compared to the standard compilation by decomposition into one-qubit and two-qubit gates. Thus a CISC quantum compiler translates into significant progress in fighting relaxation.

For demonstrating quantum CISC compilation and scalable assembly, in this paper we choose systems with linear coupling topology, i.e., qubit chains coupled by nearest-neighbour Ising interactions. The paper is organised as follows: CISC quantum compilation by optimal control will be illustrated in three different, yet typical examples

(1) the indirect $1, n$-SWAP gate,

(2) the quantum Fourier transform (QFT) ,

(3) the generalisation of the CNOT and Toffoli gate to multiply-controlled NOT gates, C^{n}NOT.

For every instance of n-qubit systems, we analyse the effects of (i) sacrificing universality by going to special instruction sets tailored to the problem, (ii) extending pair interaction gates to effective multi-qubit interaction gates, and (iii) we compare the time gain by recursive m-qubit CISC-compilation ($m \leq n$) to the two limiting cases of the standard RISC-approach ($m = 2$) on one hand and the (extrapolated) time-complexity inferred from single-CISC compliation (with $m = n$).

Preliminaries

Time Standards

When comparing times to implement unitary target gates by the RISC *vs* the CISC approach, we will assume for simplicity that local unitary operations are ‘infinitely’ fast compared to the duration of the Ising coupling evolution so that the total gate time is solely determined by the coupling evolutions unless stated otherwise. Let us emphasise, however, this stipulation only concerns the time standards. The optimal-control assisted CISC-compilation methods presented here are in no way limited to fast local controls. In particular, also the assembler step of concatenating the CISC-building blocks is independent of the ratio of times for local operations *vs* coupling interactions.

Overview on Gate and Time Complexities

For practical purposes, the complexity of a unitary quantum operation can be expressed in terms of two measures: the *gate complexity* counts the number of universal one- and two-qubit gates for exactly implementing the target operation in a circuit. Moreover, in view of fighting relaxation, we will estimate the *time complexity* in terms of consecutive time-slots with simultaneous m-qubit modules required.

In order not to raise false expectations, upon changing from universal 2-qubit decompositions (RISC) to m-qubit CISC-implementations the gate complexity for exact implemention of a *generic* n-qubit unitary operation clearly remains NP: it requires ‘exponentially many’ 2-qubit modules or m-qubit modules ($m \geq 2$) alike, yet a cut from the order of roughly $4^n/4^2$ necessary 2-qubit modules down to some $4^n/4^m$ m-qubit modules (with up to $m = 10$) is substantial and particularly valuable in few-qubit systems. More elaborate estimates will be given shortly. — Likewise, also in target modules with linear 2-qubit RISC complexity, m-qubit CISC complexity remains linear, yet when translated into time complexity it may entail sizeable speed-ups – we will show examples where they allow for accelerations by more than a factor of 13.

To be more precise, a lower bound for the number of two-qubit gates necessary to exactly implement a a generic n-qubit unitary target module was given by Barenco et al. [20]. Their parameter-counting argument is based on a gem, which deserves to be picked up for generalising it to realisations by m-qubit modules as illustrated in Fig. 2. The key is that only in the first time slot the number of parameters directly relates to the unitary group, while from the second slot onwards the parameters have to be counted in terms of *cosets* of the form $SU(2^m)/(SU(2^{m-\mu}) \otimes SU(2^\mu))$, if the m-qubit module has overlaps of μ qubits and $(m-\mu)$ qubits with the two adjacent modules in the time slot before. The number of real parameters (denoted by # for short) in the respective basic building bocks amount to

$$\#SU(2^m) \;=\; 4^m - 1 \tag{4}$$

$$\begin{aligned} \#\frac{SU(2^m)}{SU(2^{m-\mu}) \otimes \mathbb{1}_{2^\mu}} &= 4^m - 1 - (4^{m-\mu} - 1) \\ &= 4^{m-\mu}(4^\mu - 1) \end{aligned} \tag{5}$$

$$\begin{aligned} \#\frac{SU(2^m)}{SU(2^{m-\mu}) \otimes SU(2^\mu)} &= 4^m - 1 - (4^{m-\mu} - 1) - (4^\mu - 1 \\ &= (4^{m-\mu} - 1)(4^\mu - 1) \quad . \end{aligned} \tag{6}$$

With these stipulations one may readily determine the number of m-qubit gates in a unitary network of the type of Fig. 2 a, where $\frac{n}{m}$ is integer, such as to ensure to exhaust the number $4^n - 1$ of parameters of a generic n-qubit target gate to be implemented. In the first time slot there are $\frac{n}{m}$ parallel m-qubit gates (counting by the number of parameters in the group according to Eqn.4), in the second time slot there are $(\frac{n}{m} - 1)$ parallel m-qubit gates. They contribute the number of parameters of the coset (Eqn. 6), where one is forced to choose $\mu = \frac{m}{2}$ for even m and $\mu = \frac{1}{2}(m \pm 1)$ for odd m in order to be efficient. Following the same Margolus pattern one adds as many m-qubit gates (counting cosets) as required to superseed $4^n - 1$ parameters. Using Gauss’ brackets one thus obtains the number g_m of m-qubit gates needed to implement a generic n-qubit target gate

$$g_m = \left\lceil \; \frac{4^n - 1 - \frac{n}{m}(4^m - 1)}{4^m - 4^\mu - 4^{m-\mu} + 1} + \frac{n}{m} \; \right\rceil \tag{7}$$

and the respective number of time slots $t_m = \lceil \frac{g_m}{\lfloor \frac{n}{m} \rfloor} \rceil$ by

$$t_m = 1 + \left\lceil \; \frac{4^n - 1 - \frac{n}{m}(4^m - 1)}{\frac{n}{m}(4^m - 4^\mu - 4^{m-\mu} + 1)} \; \right\rceil \quad . \tag{8}$$

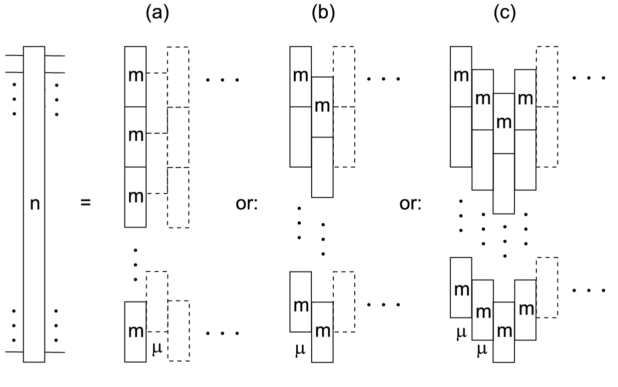

Figure 2: Decomposition of an n-qubit gate into a circuit of m-qubit gates, where m is a uniform block size and may consist of RISC modules $m = 2$ or CISC modules with $m > 2$. (a) Margolus pattern with $\frac{n}{m}$ integer, (b) $n - m\lfloor \frac{n}{m} \rfloor = \mu$ or (c) $n - m\lfloor \frac{n}{m} \rfloor = 2\mu$, so $\mu > 0$ integer.

Table I: Lower Bounds to Gate Complexities and Time Complexities for Implementing Generic n-Qubit Unitaries $SU(2^n)$

n-qubit operation with $n =$	2	3	4	5	6	7	8	9	10	20	100
no of 2-qubit gates (g_2)	1	6	27	112	453	$1,818$	$7,279$	$29,124$	$116,505$	1.22×10^{11}	1.79×10^{59}
no of 2-qubit time slots (t_2)	1	6	14	56	151	606	$1,820$	$7,281$	$23,301$	1.22×10^{10}	3.57×10^{57}
no of 10-qubit gates (g_{10})									1	$1,050,627$	1.54×10^{54}
no of 10-qubit time slots (t_{10})									1	$525,314$	1.54×10^{53}

For even n with $m = 2$ and $\mu = 1$ Eqn. 7 specialises to reproduce the result of Ref. [20], i.e. $g_2 = \frac{1}{9}(4^n - 3n - 1)$.

Next, consider Fig. 2 b and its Margolus pattern with one overhead of $\mu = n - m\lfloor\frac{n}{m}\rfloor$ qubits to be taken into account by Eqn. 5. Then the same arguments give

$$g'_m = \left\lceil \frac{4^n - 4^{m-\mu} - \lfloor\frac{n}{m}\rfloor(4^m - 1)}{4^m - 4^\mu - 4^{m-\mu} + 1} + \lfloor\frac{n}{m}\rfloor \right\rceil \tag{9}$$

$$t'_m = 1 + \left\lceil \frac{4^n - 4^{m-\mu} - \lfloor\frac{n}{m}\rfloor(4^m - 1)}{\lfloor\frac{n}{m}\rfloor(4^m - 4^\mu - 4^{m-\mu} + 1)} \right\rceil . \tag{10}$$

Finally, for a pattern with two such overheads as in Fig. 2 c, where $n - m\lfloor\frac{n}{m}\rfloor = 2\mu$, one likewise finds

$$g''_m = \left\lceil \frac{4^n + 1 - 2\cdot 4^{m-\mu} - \lfloor\frac{n}{m}\rfloor(4^m - 1)}{4^m - 4^\mu - 4^{m-\mu} + 1} + \lfloor\frac{n}{m}\rfloor \right\rceil \tag{11}$$

$$t''_m = 1 + \left\lceil \frac{4^n + 1 - 2\cdot 4^{m-\mu} - \lfloor\frac{n}{m}\rfloor(4^m - 1)}{\lfloor\frac{n}{m}\rfloor(4^m - 4^\mu - 4^{m-\mu} + 1)} \right\rceil . \tag{12}$$

With efficient implementations requiring μ to be closest to $m/2$ (*vide supra*), three overheads do not occur.

Since $g_m \geq g'_m \geq g''_m$, one may use g''_m with the most efficient setting of $\mu = \frac{1}{2}m$ for m even or $\mu = \frac{1}{2}(m-1)$ for m odd as a lower bound for the number of unitary m-qubit modules necessary to exactly implement an arbitrary generic n-qubit target unitary.

In the limit of large n, one thus obtains the bounds on gate complexities $\bar{g}_2 \simeq 4^n/9$ and $\bar{g}_{10} \simeq 4^n/1,046,529$ so $\bar{g}_{10}/\bar{g}_2 \simeq 1/116,281$. Likewise the limiting time complexities $\bar{t}_2 \simeq 2\cdot 4^n/(n\cdot 9)$ and $\bar{t}_{10} \simeq 10\cdot 4^n/(n\cdot 1,046,529)$ give a speed-up potential of $\bar{t}_{10}/\bar{t}_2 \simeq 1/23,256$ in units of the ratio of single-gate times τ_{10}/τ_2 in the respective experimental setting. These limiting speed-up ratios are nearly reached already for $n = 10$, as the numbers given in Tab. I show. In this sense, accelerations may be taken as roughly constant over the entire range of interest.

Although in generic n-qubit unitaries, the CISC speed-up may appear overwhelming, quantum algorithms are usually by construction resorting to highly non-generic unitary bulding blocks, many of which with linear complexities [20]. However, in these seemingly less rewarding yet practically relevant cases CISC compilation will turn out to be highly advantageous as demonstrated in three worked examples in the current study. — Since generic and thus highly entangled states have recently turned out to be computationally of modest use [52], recasting the above analysis in terms of 2-designs and t-designs [53, 54, 55] and following concentrations of measure will give a more realistic estimate, which is part of a different project.

Error Propagation and Relaxative Losses

As the main figure of merit we refer to a quality function

$$q := F_{\text{tr}}\; e^{-\tau/T_R} \tag{13}$$

resulting from the fidelity F_{tr} and the relaxative decay with overall relaxation rate constant T_R during a duration τ assuming independence of fidelity and decay. Moreover, for n qubits one defines as the trace fidelity of an experimental unitary module U_{exp} with respect to the target gate V_{target} the quantity

$$\begin{aligned} F_{\text{tr}} &:= \tfrac{1}{N}\;\text{Re}\;\text{tr}\{V^\dagger_{\text{target}} U_{\text{exp}}\} \\ &= 1 - \tfrac{1}{2N}\;\|V_{\text{target}} - U_{\text{exp}}\|_2^2 \quad , \end{aligned} \tag{14}$$

where both $U, V \in U(N)$ with $N := 2^n$. It follows via the simple relation to the Euclidean distance

$$\begin{aligned} \|V - U\|_2^2 &= \|U\|_2^2 + \|V\|_2^2 - 2\,\text{Re}\,\text{tr}\{V^\dagger U\} \\ &= 2N - 2N\tfrac{1}{N}\,\text{Re}\,\text{tr}\{V^\dagger U\} \\ &= 2N(1 - F_{\text{tr}}) \quad , \end{aligned}$$

the latter two identities invoking unitarity of U, V. The reason for chosing the trace fidelity is its convenient Fréchet differentiability in view of gradient-flow techniques, see also Ref. [56].

Consider an m-qubit-interaction module (CISC) with quality $q_m = F_m\; e^{-\tau_m/T_m}$ that decomposes into r universal two-qubit gates (RISC), out of which $r' \leq r$ gates have to be performed *sequentially.* Moreover, each 2-qubit gate shall be carried out with the uniform quality $q_2 = F_2\; e^{-\tau_2/T_2}$. Henceforth we assume for simplicity equal relaxation rate constants, so $T_2 = T_m$ are identified with T_R. Then, as a first useful rule of the thumb

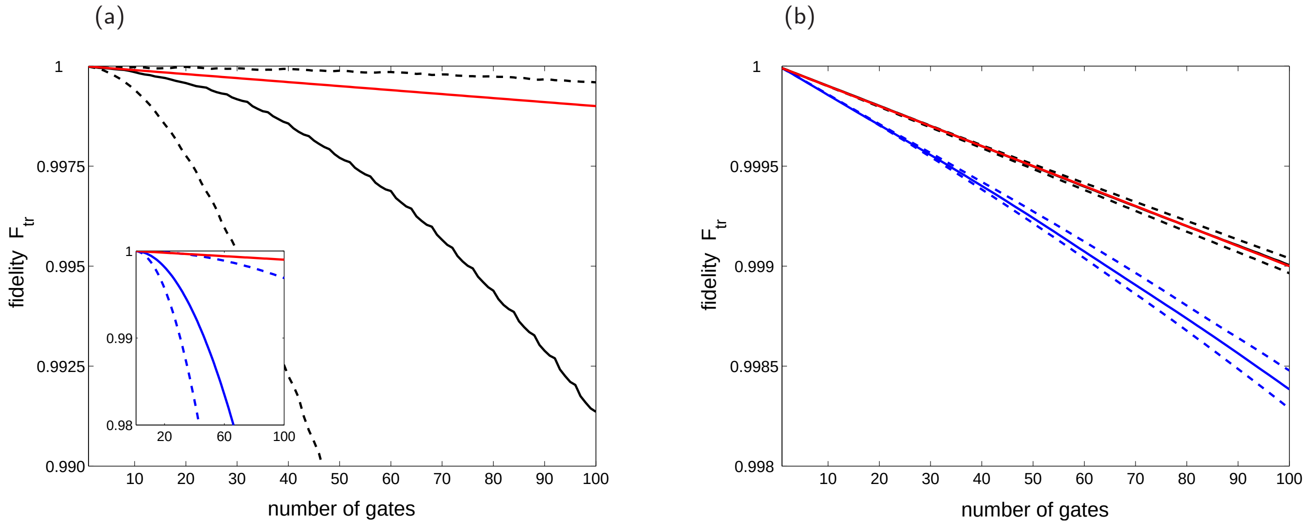

Figure 3: (Colour online) Comparison of error-propagation models for random unitary gates with $m = 2$ qubits (a) and $m = 8$ qubits (b) requiring representations with different scales. Single gate fidelity in the Monte Carlo simulations is $F_m = 0.99999$. Repetition of the same gate A (blue) is compared with repetitions of a sequence of four independent gates $ABCD$ (black). Out of 10 Monte Carlo simulations (details see text), the median (solid lines) as well as the best and worst cases (dashed lines) are given. The red solid lines denote independent error propagation $F_{\rm tr} = (F_m)^r$. Large systems ($m = 8$) with several gates ($ABCD$) resemble independent error propagation almost perfectly, as in (b) the black and the red solid lines virtually coincide.

and assuming independent error propagation, it is advantageous to compile the m-qubit module directly if $F_m > (F_2)^r$. Or more precisely taking relaxation into account, if the module can be realised with a fidelity

$$F_m > (F_2)^r \; e^{-(r\,'\cdot\tau_2-\tau_m)/T_R} \quad . \tag{15}$$

A more refined picture emerges from Monte-Carlo simulations of error propagation. To this end, compare the above independent error estimates with two scenarios for a sequence of r gates in total: (i) the r-fold repetition of single unitary gates A with individual errors meant to give A^r with $r = 1, 2, 3, \dots$ and (ii) the repetition of a sequence of four different gates A, B, C, D again each with individual errors to give $(D \circ C \circ B \circ A)^{r/4}$ where $r = 4, 8, 12, \dots$. In the sequel, we refer to case (i) as $AAAA$ and to case (ii) as $ABCD$.

For gates and errors to be generic, we use random unitaries (distributed according to the Haar measure following a recent modification [57] of the QR-algorithm). To a given random unitary m-qubit gate $A_0 \in U(2^m)$ (defining its Hamiltonian H_{A0} via $A_0 = e^{-iH_{A0}}$) we simulate a generic error as follows: from another independent unitary E_j take the matrix logarithm H_{Aj} such that $e^{-iH_{Aj}} = E_j$. Then to a given trace fidelity F, a corresponding unitary with a Monte-Carlo random error (the error being introduced on the level of the Hamiltonian generators) can readily be obtained by solving

$$\begin{aligned} F &= 1 - \frac{1}{2N}||A_0 - A_j||_2^2 \\ &= 1 - \frac{1}{2N}\big|\big|A_0 - e^{-i(H_{A0}+\delta\cdot H_{Aj})}\big|\big|_2^2 \end{aligned} \tag{16}$$

for $\delta > 0$. Along these lines one obtains the Monte-Carlo fidelities for repeating the A-gate by

$$F_{AAAA}(r) = 1 - \frac{1}{2N}\big|\big|(A_0)^r - \prod_{j=1}^{r} A_j\big|\big|_2^2 \tag{17}$$

and

$$F_{ABCD}(r) = 1-\frac{1}{2N}\big|\big|(D_0C_0B_0A_0)^{\frac{r}{4}}-\prod_{j=1}^{\frac{r}{4}}(D_jC_jB_jA_j)\big|\big|_2^2 \; , \tag{18}$$

where the product runs from right to left. These Monte-Carlo simulations are compared to the simple model of independent errors according to

$$F_{\rm ind} = (F_m)^r \quad . \tag{19}$$

As shown in Fig. 3 a, for two-qubit gates the error propagates with a vast variance, which makes it virtually unpredictable. Thus assuming independence is always too optimistic for AAAA, while for ABCD it is still mostly optimistic, although there are cases in which the errors may compensate to give less effective loss than expected under independence.

However, when moving to effective multi-qubit gates, i.e., CISC modules, the generic situation becomes more predictable. For example, in 8-qubit random unitary gates, Fig. 3 b shows that AAAA is significantly deviating from independent error propagation, whereas ABCD resembles independent error propagation almost perfectly. The situation is qualitatively exactly the same even if the

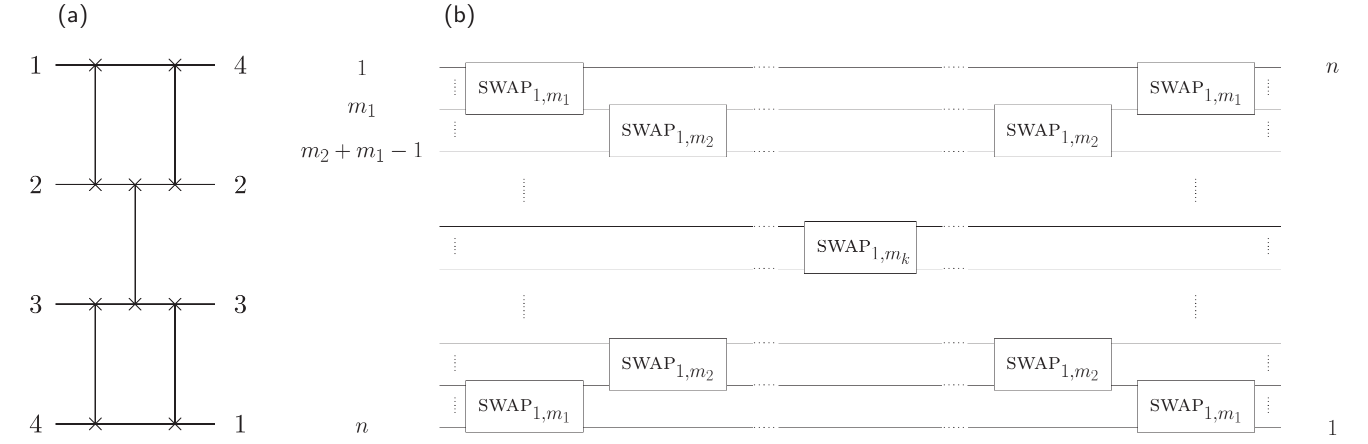

Figure 4: (a) Simple starting point: building a SWAP$_{1,4}$ gate from five SWAPs$_{1,2}$. (b) Generalisation: assembling a SWAP$_{1,n}$ by four SWAPs$_{1,m_j}$ for each type $j = 1, 2, \ldots, k-1$ and one single SWAP$_{1,m_k}$ so that $m_k + 2\sum_{j=1}^{k-1}(m_j - 1) = n$.

single gate error is larger as tested by analogous Monte-Carlo simulations setting $F_{\text{tr}} = 0.99$ or $F_{\text{tr}} = 0.96$ (not shown).

In the sequel, we will—for the sake of simplicity—often assume independent error propagation at the expense of systematically underestimating the pros of CISC compilation compared to the standard RISC compilation into universal local and two-qubit gates.

Computational Methods and Devices

Following the lines of our previous work on time complexity [34], we used the GRAPE algorithm [45] for direct CISC compilation. It tracks the fixed final times down to the shortest durations of controls still allowing for synthesising the unitary target gates with full fidelity. This gives currently the best known upper bounds to the minimal times required to realise a target module on a concrete hardware setting. We extended our parallelised C++ code of the GRAPE package described in [47] by adding more flexibility allowing to efficiently exploit available parallel nodes independent of internal parameters [58]. Moreover, faster algorithms for matrix exponentials on high-dimensional systems based on approximations by Tchebychev series have been developed [59] specifically in view of application to large quantum systems [58]. Thus computations could be performed on the HLRB-II supercomputer cluster at *Leibniz Rechenzentrum* of the Bavarian Academy of Sciences Munich. It provides an SGI Altix 4700 platform equipped with 9728 Intel Itanium2 Montecito Dual Core processors with a clock rate of 1.6 GHz, which give a total LINPACK performance of 63.3 TFlops/s. The present explorative study exploited the time allowance of approx. 500.000 CPU hours.

I. THE $1, n$ SWAP OPERATION

The easiest and most basic examples to illustrate the pertinent effects of optimal-control based CISC-quantum compilation are the respective indirect SWAP$_{1,n}$ gates in spin chains of n qubits coupled by nearest-neighbour Ising interactions with J_{ZZ} denoting the coupling constant.

For the SWAP$_{1,2}$ unit there is a standard textbook decomposition into three CNOTs. Thus for Ising-coupled systems and in the limit of fast local controls, the total time required for an SWAP$_{1,2}$ is $3/(2J_{ZZ})$, and there is no faster implementation [60, 61]. Note, however, that in systems coupled by the isotropic Heisenberg interaction XXX, the SWAP$_{1,2}$ may be directly implemented just by letting the system evolve for a time of only $1/(2J_{XXX})$. Sacrificing universality, it may thus be advantageous to regard the SWAP$_{1,2}$ as basic unit for the SWAP$_{1,n}$ task rather than the universal CNOT. Clearly, any even-order SWAP$_{1,2n}$ can be built from SWAPs$_{1,2}$ along the lines of the most obvious scheme of Fig. 4 a. (The odd-order SWAPs$_{1,2n-1}$ follow, e.g., from SWAP$_{1,2n}$ by omitting qubit $2n$ and all the gates connected to it.)

Moreover, the generalisation to decomposing a SWAP$_{1,n}$ into a sequence with k different SWAP$_{1,m_j}$ building blocks (where $j = 1, 2, \ldots, k$) as shown in Fig. 4 b is straightforward by ensuring $m_k + 2\sum_{j=1}^{k-1}(m_j - 1) = n$. Due to its symmetry, the total duration then amounts to

$$\tau(\text{SWAP}_{1,n}) = \tau(\text{SWAP}_{1,m_k}) + 2\sum_{j=1}^{k-1} \tau(\text{SWAP}_{1,m_j}) \qquad (20)$$

and the overall quality as a function of the fidelities of the constitutent gates reads

$$q_{\text{SWAP}_{1,n}} = F(\text{SWAP}_{1,m_k}) \prod_{j=1}^{k-1} F(\text{SWAP}_{1,m_j})^4 \times e^{-\tau(\text{SWAP}_{1,n})/T_R} \quad . \qquad (21)$$

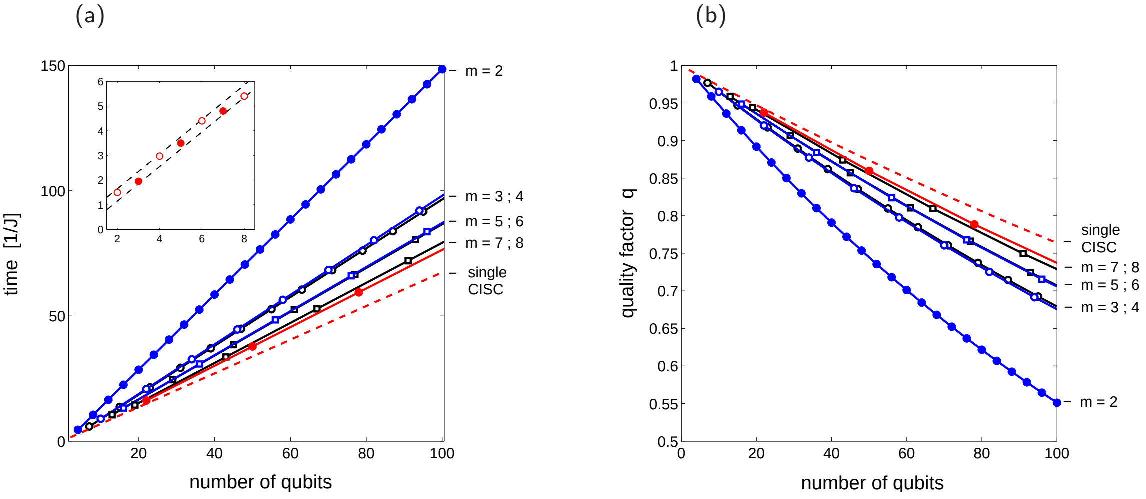

Figure 5: (Colour online) (a): Times required for indirect SWAPs$_{1,n}$ on linear chains of n Ising-coupled qubits by assembling SWAP$_{1,m}$ building blocks reaching from $m = 2$ (RISC) up to $m = 8$ (CISC). Using linear regression, the dashed line is an extrapolation of the direct single-CISC compilations shown in the inset to large number of qubits, where direct CISC compilation is virtually impossible on classical computers. Time units are expressed as $1/J_{ZZ}$ assuming the duration of local operations can be neglected compared to coupling evolutions (details in the text). (b): Translation of the effective gate times into overall quality figures $q = (q_m)^{r_m}$ for an effective gate assembled from r_m components of single qualities $q_m := F_m \, e^{-\tau_m/T_R}$ (with the respective component fidelities homogeneously falling into a narrow interval $F_m \in [0.99994, 0.99999]$ for $m = 3, \ldots, 8$). Data are shown for a uniform relaxation rate constant of $1/T_R = 0.004 J_{ZZ}$.

Now, the SWAP$_{1,m_j}$ building blocks themselves can be precompiled into time-optimised single complex instruction sets by exploiting the GRAPE-algorithm of optimal control up to the current limits of m_j imposed by CPU-time allowance.

Proceeding in the next step to large n, Fig. 5 underscores how the time required for SWAP$_{1,n}$ gates decreases significantly by assembling precompiled SWAP$_{1,m_j}$ building blocks as CISC units recursively up to a multi-qubit interaction size of $m_j = 8$, where the speed-up is by a factor of nearly 2. Clearly, such a set of SWAP$_{1,m_j}$ building blocks with $m_j \in \{2, 3, 4, 5, 6, 7, 8\}$ allows for efficiently synthesising any SWAP$_{1,n}$. Assuming for the moment that a linear time complexity of the SWAP$_{1,n}$ can be taken for granted, one may extrapolate the results of direct CISC compilation from the range of the inset of Fig. 5 a to a large number of qubits. One thus obtains an estimated upper limit to the time complexity of the SWAP$_{1,n}$. This is indicated by the dashed line, the slope of which will be defined as Δ_∞. Likewise, the irrespective slopes of the m-qubit decomposition are denoted by Δ_m.

With these stipulations, we introduce as a measure for the potential of CISC compilation (versus RISC compilation) the ratio of the slopes

$$\pi_{\text{CISC}} := \frac{\Delta_2}{\Delta_\infty} \tag{22}$$

and as a measure for the extent to which this potential has been exhausted by m-qubit CISC compilation the ratio

$$\eta_m := \frac{\Delta_\infty}{\Delta_m} \tag{23}$$

thus providing as convenient measure of improvement

$$\xi_m := \frac{\Delta_2}{\Delta_m} = \eta_m \cdot \pi_{\text{CISC}} \quad . \tag{24}$$

The data of Fig. 5 thus give a potential of $\pi_{\text{CISC}} = 2.16$; by $m = 8$-qubit interactions it is already pretty well exhausted, as inferred from $\eta_8 = 0.87$. The current CISC over RISC improvement then amounts to $\xi_8 = 1.88$.

On the other hand, deducing from Fig. 5 right away that the time complexity of SWAPs$_{1,n}$ ought to be linear would be premature: although the slopes seem to converge to a non-zero limit, numerical optimal control may become systematically inefficient for larger interaction sizes m. Therefore, although improbable, e.g., convergence of the slopes to a value of zero cannot be safely excluded on the current basis of findings. This also means a logarithmic time complexity can ultimately not be excluded either.

Summarising the results for the indirect SWAPs in terms of the three criteria described in the introduction, we have the following: (i) in Ising coupled qubit chains, there is no speed-up by changing the basic unit from the universal CNOT into a SWAP$_{1,2}$, whereas in isotropically coupled systems the speed-up amounts to a factor of three; (ii) extending the building blocks of SWAP$_{1,m}$ from $m = 2$ (RISC) to $m = 8$ (CISC) gives a speed-up by a factor of nearly two under Ising-type couplings; (iii) the numerical data are consistent with a time complexity converging to a linear limit for the SWAP$_{1,n}$ task in Ising chains, however, there is no proof for it yet.

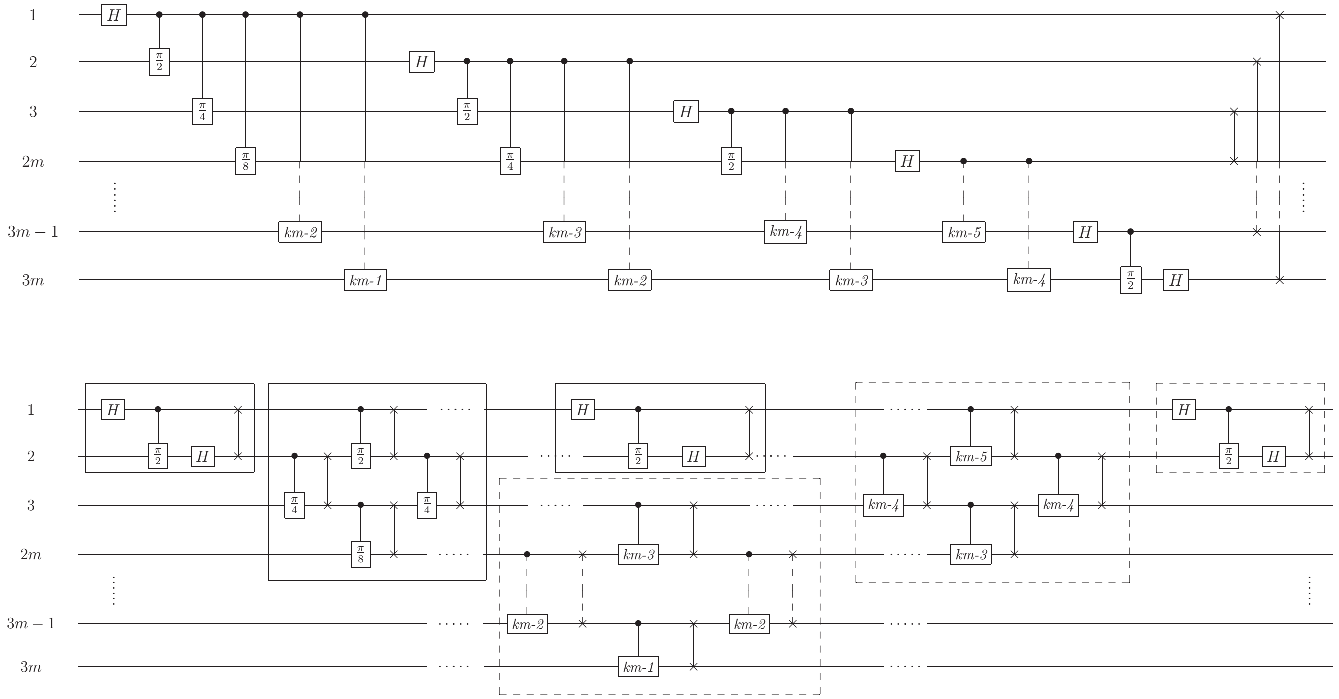

Figure 6: By rearranging the SWAPs and controlled phase gates, the standard decomposition of a $3m$-qubit quantum Fourier transform, QFT (top trace) reduces to a realisation adapted to a coupling topology of linear nearest-neighbour interactions (lower trace) with a $2m$-qubit QFT, m-qubit cP-SWAPs (solid boxes), and an m-qubit QFT (dashed box). The notation $(km - \nu)$ is a shorthand for a rotation angle of $\phi = \frac{\pi}{2^{km-\nu}}$.

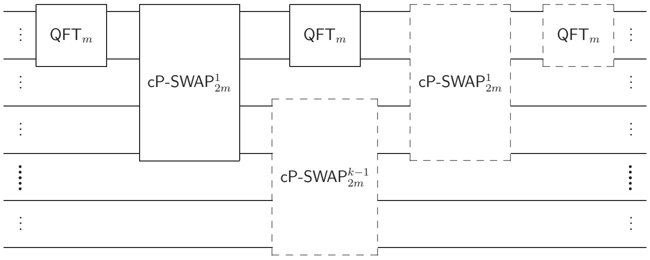
QFT_m
$\mathsf{cP\text{-}SWAP}^1_{2m}$
QFT_m
$\mathsf{cP\text{-}SWAP}^{k-1}_{2m}$
$\mathsf{cP\text{-}SWAP}^1_{2m}$
QFT_m

Figure 7: For $k \geq 2$, a (km)-qubit QFT can be assembled from k times an m-qubit QFT and $\binom{k}{2}$ instances of $2m$-qubit modules cP-SWAP$^j_{2m}$, where the index j of different phase-rotation angles takes the values $j = 1, 2, \ldots, k-1$. The dashed boxes correspond to Fig. 6 and show the induction $k \mapsto k+1$.

II. THE QUANTUM FOURIER TRANSFORM (QFT)

Since many quantum algorithms take advantage of efficiently solving some hidden subgroup problem, the quantum Fourier transform plays a central role [6, 7, 8].

In order to realise a QFT on large qubit systems, our approach is the following: given an m-qubit QFT, we show that for obtaining a $(k \cdot m)$-qubit QFT by recursively using multi-qubit building blocks, a second type of module is required, to wit a combination of controlled phase gates and SWAPs, which henceforth we dub m'-qubit cP-SWAP for short.

Here we present two alternatives: variant I with $m' = 2m$ and, as a special case, variant II for even $m' = m$.

Choosing $m = 2$ and $k = 3$ for a start, the recursive construction is illustrated in Fig. 6. The top trace shows the standard textbook realisation of a 6-qubit QFT. By shifting the final SWAP operations, it can be rearranged into the sequence of gates depicted in the lower trace. Note that the gates appearing in solid boxes constitute a $2m$-qubit QFT (which itself is made of two m-qubit QFTs and a central m-qubit cP-SWAP), while the ones in dashed boxes have to be added for a $3m$-qubit QFT. For $m = 2$ we have thus shown how a $3m$-qubit QFT reduces to a $2m$-qubit QFT, two $2m$-qubit cP-SWAPs, and an m-qubit QFT. So with $2m$ providing a foundation, at the same time we have also illustrated the induction from a $k \cdot m$-QFT to a $(k+1) \cdot m$-QFT. Moreover, the same construction principle holds for any block size $m = 2, 3, \ldots$, which can readily be proven by a straightforward, but lengthy induction from m to $m + 1$.

One thus arrives at the desired block decomposition of a general $(k \cdot m)$-qubit QFT as shown in Fig. 7 (which is variant I; the less effective variant II can be found in Appendix A): it requires k times the same m-qubit QFT interdispersed with $\binom{k}{2}$ times an $2m$-qubit cP-SWAP, out of which $k - 1$ show different phase-roation angles. For all m and $j = 1, 2, \ldots, (k - 1)$, one finds the following observations:

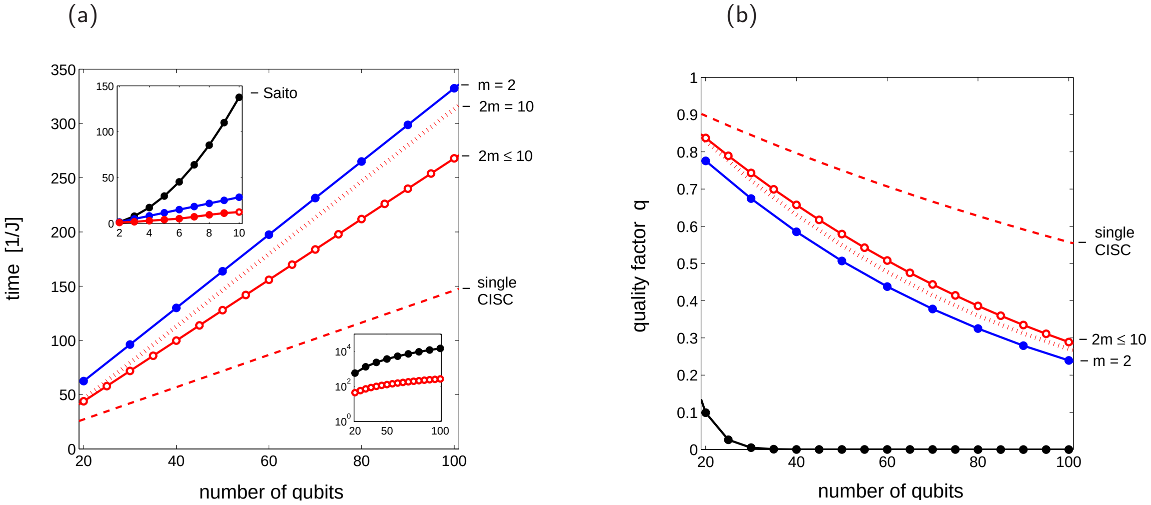

Figure 8: (Colour online) Comparison of CISC-compiled QFT (red) with standard RISC compilations ($m = 2$) following the scheme by Saito [62] (black) or Blais [25] (blue): (a) times for implementation translate into quality factors (b) for a relaxation rate constant of $1/T_R = 0.004 J_{ZZ}$. Again, the dashed red line extrapolates from the direct single-CISC compilations shown in the upper inset of (a), the lower inset giving in logarithmic scales the times needed for the standard textbook RISC compilation ($m = 2$) on a linear Ising chain. Dotted red lines represent the less favourable results from QFT variant II (Appendix A).

1. a cP-SWAP$^j_{2m}$ takes as least as long as a QFT$_m$;
2. a QFT$_m$ takes as least as long as a cP-SWAP^j_m;
3. a cP-SWAP^j_m takes least as long as a cP-SWAP$^{j+1}_m$.

Thus the duration of a $(k \cdot m)$-qubit QFT built from m-qubit and $2m$-qubit modules amounts to

$$\begin{aligned}\tau(\text{QFT}_{k\cdot m}) = 2{\cdot}\tau(\text{QFT}_m) + (k-1)\cdot\tau(\text{cP-SWAP}^1_{2m})\\ + (k-2)\cdot\tau(\text{cP-SWAP}^2_{2m}) \quad .\end{aligned} \tag{25}$$

Next, consider the overall quality of a $(k \cdot m)$-qubit QFT in terms of its two types of building blocks, namely the basic m-qubit QFT as well as the constituent $2m$-qubit cP-SWAPs with their respective different rotation angles. It reads

$$\begin{aligned} q_{\text{QFT}_{k\cdot m}} \;=\; (F_{\text{QFT}_m})^k \,\Big(\prod_{j=1}^{k-1} (F_{\text{cP-SWAP}^j_{2m}})^{k-j}\,\Big)\\ \times\; e^{-(\tau_{\text{QFT}_{k\cdot m}}/T_R)} \;. \end{aligned} \tag{26}$$

In the following, we will neglect rotations as soon as their angle falls short of a threshold of $\pi/2^{10}$. This approximation is safe since it is based on a calculation of a 20-qubit QFT, where the truncation does not introduce any relative error beyond 10^{-5}. According to the block decomposition of Fig. 7, thus three instances of cP-SWAPs are left, since all cP-SWAP$^j_{10}$ elements with $j \geq 3$ boil down to mere SWAP gates due to truncation of small rotation angles. The representation of these cP-SWAP modules is shown in Appendix B as Fig. 17.

With these stipulations, we address the task of assembling an $(k{\cdot}10)$-qubit QFT, exploiting the limits of current allowances on the HLRB-II cluster. This translates into using 10-qubit cP-SWAP building blocks ($2m = 10$) and the 5-qubit QFT ($m = 5$) in the sense of a $(2k \cdot 5)$-qubit QFT. Its duration $\tau(\text{QFT}_{2k\cdot 5})$ is readily obtained as in Eqn. 25 thus giving an overall quality of

$$\begin{aligned} q_{\text{QFT}_{2k\cdot 5}} = (F_{\text{QFT}_5})^{2k}\,(F_{\text{cP-SWAP}^1_{10}})^{2k-1}\,(F_{\text{cP-SWAP}^2_{10}})^{2k-2}\\ \times\;\; (F_{\text{cP-SWAP}^3_{10}})^{\binom{2k}{2}-4k+3} \quad e^{-\tau_{\text{QFT}_{2k\cdot 5}}/T_R}\;. \end{aligned} \tag{27}$$

Based on this relation, the numerical results of Fig. 8 show that a CISC-compiled QFT is moderately superior to the standard RISC versions [25, 62]. Although the potential of CISC compilation amounts to $\pi_{\text{CISC}} = 2.27$, recursively assembling 5-qubit QFTs and 10-qubit cP-SWAPs only exploits about half of it as apparent in the value of $\eta_{5,10} = 0.53$.

As has been pointed out by Zeier [63], the decomposition of a many-qubit QFT into smaller QFTs and concatenations of a permutation matrix and a diagonal matrix roots back in a principle already used in the Cooley-Tukey algorithm [9] for the discrete Fourier transform (DFT): Let $N = m \cdot q$. Then one obtains [64, 65]

$$\begin{aligned}\text{DFT}_N &= L \circ (\text{DFT}_m \otimes \mathbb{1}_q) \circ D \circ (\mathbb{1}_m \otimes \text{DFT}_q)\\ &= (\mathbb{1}_q \otimes \text{DFT}_m) \circ (L \circ D) \circ (\mathbb{1}_m \otimes \text{DFT}_q)\;,\end{aligned} \tag{28}$$

where $L \in \text{Mat}_N$ is a permutation matrix. Moreover, setting $\omega := e^{2\pi i/N}$, the diagonal matrix takes the form

$$D = \text{diag}\,(\omega^{t_k} | t_k = (k \bmod m)\lfloor \tfrac{k}{m} \rfloor \text{ for } k = 0, 1, 2, \ldots N{-}1) \tag{29}$$

Therefore, the QFT decompositions made use of here exactly follow the classical scheme in the second line of

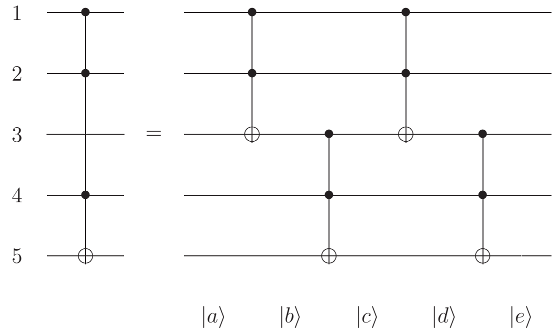

Figure 9: Decomposition of a C^3-NOT with one ancilla qubit into four C^2-NOTs (Toffoli gates) according to ref. [20]. States $|a\rangle$ through $|e\rangle$ are explained in the text.

Eqn. 28, the expression $(L \circ D)$ corresponding to the cP-SWAP.

III. THE MULTIPLE-CONTROLLED NOT GATE (C^{n}NOT)

Multiply-controlled CNOT gates generalise Toffoli's gate. Here, we move from C^{2}NOT to C^{n-2}NOT in an n-qubit system with one ancilla and one target qubit. The reason for the ancilla qubit being that it turns the problem to linear complexity [20]. Moreover, in view of realistic large systems, we assume again a topology of a linear chain coupled by nearest-neighbour Ising interactions. Since C^{m}NOT-gates frequently occur in error-correction schemes, they are highly relevant in practice.

Here we address the task of decomposing a C^{n-2}NOT into lower CNOTs and indirect SWAP gates (see Sec. I).

To this end, we will generalise the basic principle of reducing a C^{n}NOT to C^{m}NOT gates with $m < n$ that can be demonstrated by decomposing a C^{3}NOT into Toffoli gates according to scheme of Fig. 9 devised by Barenco *et al.* in [20]. Starting with any of the 2^5 computational basis states $|x_1, x_2, x_3, x_4, x_5\rangle$ (where $x_k \in \{0,1\}$, $\oplus$ denotes addition mod 2, and $x_k x_\ell$ being the usual scalar product) track the effect of the gate sequentially from state $|a\rangle$ through state $|e\rangle$

$$|a\rangle = |x_1, x_2, x_3, x_4, x_5\rangle$$

$$|b\rangle = |x_1, x_2, x_3 \oplus x_1x_2, x_4, x_5\rangle$$

$$\begin{aligned} |c\rangle &= |x_1, x_2, x_3 \oplus x_1x_2, x_4, x_5 \oplus x_4(x_3 \oplus x_1x_2)\rangle \\ &= |x_1, x_2, x_3 \oplus x_1x_2, x_4, x_5 \oplus x_4x_3 \oplus x_1x_2x_4\rangle \end{aligned}$$

$$\begin{aligned} |d\rangle &= |x_1, x_2, x_3 \oplus x_1x_2 \oplus x_1x_2, x_4, x_5 \oplus x_4x_3 \oplus x_1x_2x_4\rangle \\ &= |x_1, x_2, x_3, x_4, x_5 \oplus x_4x_3 \oplus x_1x_2x_4\rangle \end{aligned}$$

$$\begin{aligned} |e\rangle &= |x_1, x_2, x_3, x_4, x_5 \oplus x_4x_3 \oplus x_4x_3 \oplus x_1x_2x_4\rangle \\ &= |x_1, x_2, x_3, x_4, x_5 \oplus x_1x_2x_4\rangle \end{aligned}$$

to see the overall effect of the gate sequence is a C^{3}NOT thus proving the decomposition.

Fig. 10 provides a generalisation of the scheme in Fig. 9: in the first place (a), the number of control qubits is reduced by introducing k blocks with m_3 qubits that are left invariant. The price for this reduction is a fourfold occurence of the reduced building blocks. In the second step (b), the reduced building blocks are expanded into a sequence with two central C^{m_2}NOTs, two terminal C$^{(m_3+1)}$NOTs and two lots of $2(k-1)$ times C$^{(m_3+1)}$NOT each. For $k = 4, 5, 6, \ldots$ part (a) and (b) can be expanded in a general concatenated way thus entailing an overall duration of

$$\begin{aligned} \tau(\mathrm{C}^{n-2}\mathrm{NOT})\Big|_{k\geq 4} &= 4\tau(\mathrm{C}^{m_1}\mathrm{NOT}) \\ &+ 4\tau(\mathrm{C}^{m_2}\mathrm{NOT}) + \tau(\mathrm{SWAP}_{1,m_2}) \\ &+ (13k-8)\tau(\mathrm{C}^{m_3+1}\mathrm{NOT}) \\ &+ (1-\delta_{m_3,1})\;(13k+3)\tau(\mathrm{SWAP}_{1,m_3})\;. \end{aligned} \tag{30}$$

For completeness, note that the cases $k = 3, 2, 1$ have to be treated separately, since they only allow for less and less densly concatenated expansions (not shown). Their respective durations are

$$\begin{aligned} \tau(\mathrm{C}^{n-2}\mathrm{NOT})\Big|_{k=1} &= 4\tau(\mathrm{C}^{m_1}\mathrm{NOT}) + 2\tau(\mathrm{SWAP}_{1,m_1+1}) \\ &+ 4\tau(\mathrm{C}^{m_2}\mathrm{NOT}) + 2\tau(\mathrm{SWAP}_{1,m_2}) \\ &+ 8\tau(\mathrm{C}^{m_3+1}\mathrm{NOT}) + (1-\delta_{m_3,1})\;16\tau(\mathrm{SWAP}_{1,m_3}) \end{aligned} \tag{31}$$

$$\begin{aligned} \tau(\mathrm{C}^{n-2}\mathrm{NOT})\Big|_{k=2} &= 4\tau(\mathrm{C}^{m_1}\mathrm{NOT}) \\ &+ 4\tau(\mathrm{C}^{m_2}\mathrm{NOT}) + \tau(\mathrm{SWAP}_{1,m_2}) \\ &+ 24\tau(\mathrm{C}^{m_3+1}\mathrm{NOT}) + (1-\delta_{m_3,1})\;32\tau(\mathrm{SWAP}_{1,m_3}) \end{aligned} \tag{32}$$

$$\begin{aligned} \tau(\mathrm{C}^{n-2}\mathrm{NOT})\Big|_{k=3} &= 4\tau(\mathrm{C}^{m_1}\mathrm{NOT}) \\ &+ 4\tau(\mathrm{C}^{m_2}\mathrm{NOT}) + \tau(\mathrm{SWAP}_{1,m_2}) \\ &+ 37\tau(\mathrm{C}^{m_3+1}\mathrm{NOT}) + (1-\delta_{m_3,1})\;48\tau(\mathrm{SWAP}_{1,m_3})\;. \end{aligned} \tag{33}$$

However, the total number of gates only depends on $k = 1, 2, 3, \ldots$, so that obtains as the overall quality

$$\begin{aligned} q\Big|_k &= (F_{C^{m_1}NOT})^4 (F_{\mathrm{SWAP}_{1,m_1+1}})^4 \\ &\times (F_{C^{m_2}NOT})^4 (F_{\mathrm{SWAP}_{1,m_2}})^2 \\ &\times (F_{C^{m_3+1}NOT})^{16k-8}\;(F_{\mathrm{SWAP}_{1,m_3}})^{(1-\delta_{m_3,1})\,16k} \\ &\times e^{-\tau(C^{n-2}NOT)\big|_k/T_R}\;. \end{aligned} \tag{34}$$

Given the duration of the decomposition as in Eqn. 30, it is easy to see that implementing the m_1 control qubits

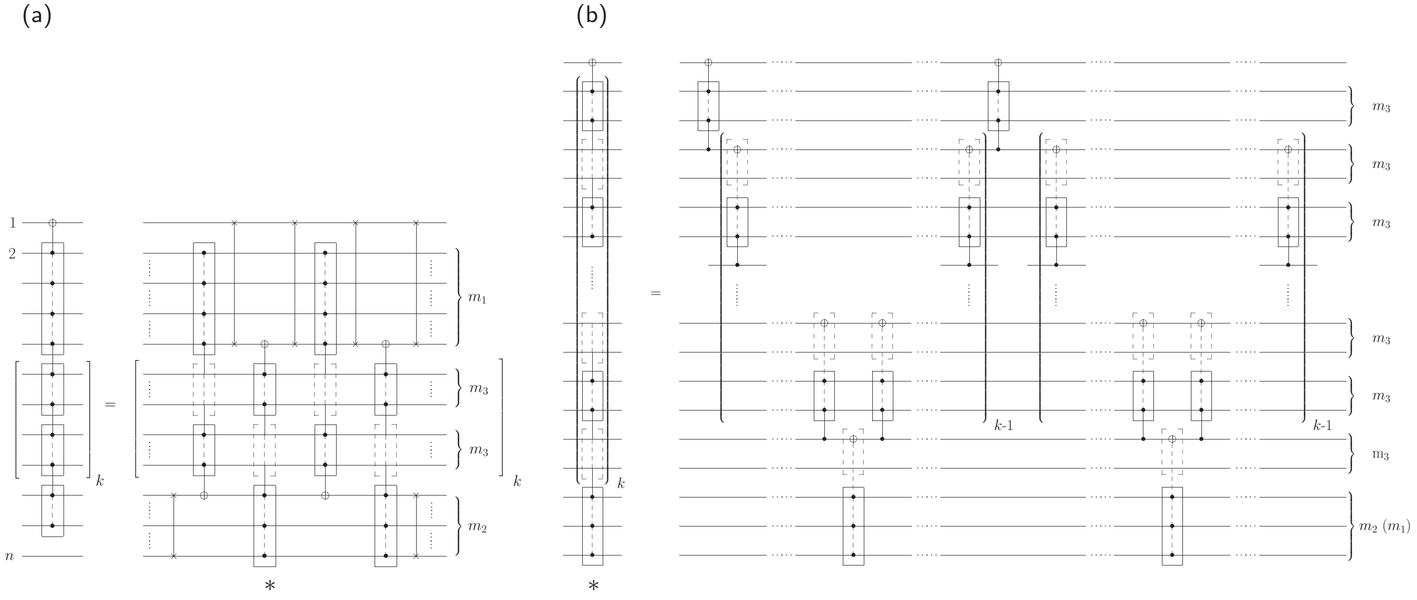

Figure 10: Decomposition of a C^{n-2}NOT gate on a linear coupling topology: (a) reduction of the number of control qubits to four intermediate gates with fewer control qubits and (b) decomposition of the intermediate multiply-controlled NOT-gate appearing in (a). In an n-qubit system, there is one target qubit, one ancilla qubit and $n-2$ control qubits; so $m_1+(m_2-1)+2km_3 = n-2$ with $m_1, m_3 \geq 1$ and $m_2 \geq 2$. Read the brackets $[\cdot]_k$ in (a) as *to be expanded k times* and $(\cdot)_k$ in (b) as *expanded k-fold*.

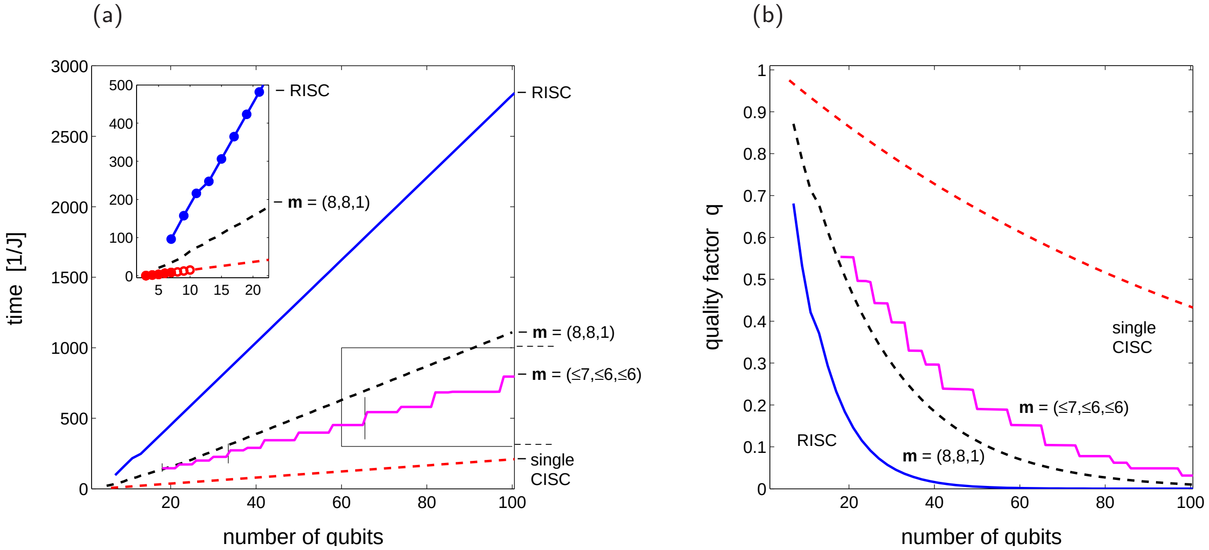

Figure 11: (Colour online) Comparing implementations of C^{n-2}NOTs on a linear Ising spin chain using CNOT and $\text{SWAP}_{1,2}$ modules for the RISC compilation or multi-qubit building blocks according to the CISC assembler scheme of Fig. 10. As a short-hand, the different numbers of control qubits are expressed by $\mathbf{m} := (m_1, m_2, m_3)$. Using the expansion of Fig. 10, the CISC results (black solid lines) are obtained for $k = 4, 5, 6, \ldots$ with $m_1 = 8$ and $m_2 = 8$ (for odd n) or $m_2 = 7$ (for even n), while $m_3 = 1$ thus ensuring $m_1 + (m_2 - 1) + 2km_3 = n - 2$. The red dotted line extrapolates again the direct CISC results beyond 10 qubits. In (a) deviations from straight lines occur, as the cases $k = 1, 2, 3$ follow special concatenation patterns (see text), while $k = 4, 5, 6, \ldots$ are generic. The inset in (a) also shows results of a non-scalable recursive expansion that is confined up to 19 qubits (blue circles). The step functions with periods indicated by tags represent a faster alternative explained in the next section, where the boxed part of trace (a) is blown up in Fig. 15.

comes with the lowest time weight (4) and without a time overhead of auxiliary gates. Implementing the m_2 control qubits, however, requires the same time weight (4), but entails the time for one auxiliary SWAP_{1,m_2} gate. In order to implement the $k \cdot m_3$ control qubits, in turn, a sizeable amount of auxiliary SWAPs are needed.

Therefore, whenever high fidelities can be reached (so that the quality is limited by relaxation not by fidelity), a good strategy of combining the expansive decomposition in Fig. 10 a with the recursive decomposition in part (b) is the following: given $n-2$ control qubits and with the current limitation from direct CISC compilation being $m_j \leq 9$, choose m_1 to be the largest, m_2 to be the second largest and such that one obtains an even number for $n - m_1 - m_2 - 1 = 2km_3$.

In the next step, a decision has to be made in order to minimise the contributions in the last two lines of Eqn. 30, whenever there are several integer solutions $km_3 = k'm'_3$. So for integer $k \geq 4$, this amounts to the ordinary minimisation task

$$\begin{aligned} \min_k \, (13k-8)\left\{(m_3+2)\Delta_C + a\right\} \\ + (1-\delta_{m_3,1})\,(13k+3)\left\{m_3\Delta_S + b\right\} \\ \text{subject to:} \quad km_3 = \text{const.} \equiv \frac{n-(m_1+m_2+1)}{2} \end{aligned} \tag{35}$$

Here we approximate the times for a C$^{m_3+1}$NOT by the linear expression $\tau(\text{C}^{m_3+1}\text{NOT}) = (m_3+2)\Delta_C + a$ and likewise for the SWAP$_{1,m_3}$ by $\tau(\text{SWAP}_{1,m_3}) = m_3\Delta_S + b$ with the values for the slopes Δ_C, Δ_S and the offsets a and b being taken from the respective linear regression for extrapolating Δ_∞ for direct CISC compilation ($\Delta_C^{(\infty)} = 2.15$ and $\Delta_S^{(\infty)} = 0.69$ as well as $a = -4.48$ and $b = 0.06$). In the setting of these parameters, Eqn. 30 implies it is timewise advantageous to choose as the decomposition of the interior block in Fig. 10 b the counter-intuitive option with a large number k of small block sizes m_3. This is because in the above parameter setting, the duration takes its minimum on the margin circumventing the time overhead skipped by $(1-\delta_{m_3,1}) = 0$ thus giving high repetitions $k = \frac{n-(m_1+m_2+1)}{2}$ and smallest block sizes $m_3 = 1$ corresponding to Toffoli gates. The speed-up is illustrated in Fig. 11: although it amounts to a factor of 2.45 compared to the standard RISC decomposition, the potential as extrapolated from direct CISC compilation up to nine qubits gives a lower bound for the speed-up by 13.6.

Faster Alternatives of C^{n-2}NOT

Since the potential of CISC compiling C^{n}NOT gates is largely not yet exploited by the previous scheme, it is worth showing a faster scalable decomposition at the expense of being more elaborate. To this end, we proceed in two steps, first we show the general principle of an auxiliary backbone gate, namely an indirect CNOT between qubit 1 and some distant qubit $\ell+1$ (which may be separated by ℓ intermediate qubits, e.g., in a linear coupling topology). Second we implement the resulting faster alternative into Fig. 10 b.

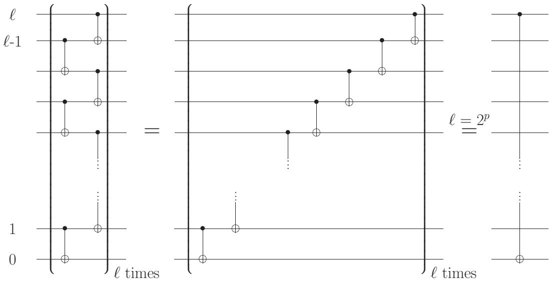

Figure 12: Principle of generating an indirect CNOT by restricting $\ell = 2^p$ as described in the text.

Fig. 12 shows the principle identities: if ℓ is an integer power of two, the two identities hold for any $\ell = 2^p$. The second identity is easy to see since the ascending series of CNOT gates can be represented by a Jordan matrix over the field of binary numbers $\mathbb{Z}_2^{\ell+1} := \{0,1\}^{\ell+1}$ with the addition modulo 2 so as to take the form

$$(J)_{\mathbb{Z}_2^{\ell+1}} = \begin{pmatrix} 1 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \cdots & 0 & 0 & 0 \\ \vdots & & & & \ddots & & & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 1 \end{pmatrix}^t . \tag{36}$$

In terms of natural numbers, its ℓ^{th} power reads

$$J^\ell = \begin{pmatrix} 1 & \binom{\ell}{1} & \binom{\ell}{2} & \binom{\ell}{3} & \cdots & \binom{\ell}{\ell-2} & \binom{\ell}{\ell-1} & \binom{\ell}{\ell} \\ 0 & 1 & \binom{\ell}{1} & \binom{\ell}{2} & \cdots & \binom{\ell}{\ell-3} & \binom{\ell}{\ell-2} & \binom{\ell}{\ell-1} \\ 0 & 0 & 1 & \binom{\ell}{1} & \cdots & \binom{\ell}{\ell-4} & \binom{\ell}{\ell-3} & \binom{\ell}{\ell-2} \\ 0 & 0 & 0 & 1 & \cdots & \binom{\ell}{\ell-5} & \binom{\ell}{\ell-4} & \binom{\ell}{\ell-3} \\ \vdots & & & & \ddots & & & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 & \binom{\ell}{1} & \binom{\ell}{2} \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 & \binom{\ell}{1} \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 1 \end{pmatrix}^t . \tag{37}$$

For $\ell = 2^p$ with $p = 1, 2, 3, 4, \ldots$ it gives the desired indirect C$^{1,\ell+1}$NOT as seen in the representation over $\mathbb{Z}_2^{\ell+1}$

$$(J^\ell)_{\mathbb{Z}_2^{\ell+1}} = \begin{pmatrix} 1 & 0 & 0 & 0 & \cdots & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \cdots & 0 & 0 & 0 \\ \vdots & & & & \ddots & & & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 1 \end{pmatrix}^t = (\text{C}^{1,\ell+1}\text{NOT})^t_{\mathbb{Z}_2^{\ell+1}} . \tag{38}$$

This is because due to a theorem by Lucas [66] only for ℓ being an integer power of two, all the binomial coefficients

$\binom{\ell}{j}$ with $j = 1, 2, \ldots, (\ell-1)$ are even, while $\binom{\ell}{0} = \binom{\ell}{\ell} = 1$. They are therefore the only ones not to vanish in the representation over $\mathbb{Z}_2^{\ell+1}$.

The principle backbone summerised in the identities of Fig. 12 may then be extended for more general purposes: (1) Without changing ℓ one may insert further control qubits in the sense of replacing any CNOT by a Toffoli or a higher C^mNOT. (2) Likewise, one may formally insert further target qubits to be flipped so that the NOT component is performed on more than one qubit. These two extensions enable a faster alternative for decomposing the module of Fig. 10 a than given in Fig. 10 b. This alternative is shown in Fig. 13 a. Due to the backbone scheme of Fig. 12, the only constraint is that '1 plus the number of neutral blocks of size $m_{3,j}$ (represented by dashed boxes in Fig. 13 a) equals $\ell = 2^p$ with $p \in \mathbb{N}$'. Changing the assembly of a C^{n-2}NOT from the scheme of Fig. 10 to the alternative of Fig. 13 follows by identifying $k = \ell - 1 = 2^p - 1$, i.e.

$$\begin{aligned} n - 2 &= m_1 + (m_2 - 1) + 2km_3 \\ &= m_1 + (m_2 - 1) + \sum_{j=1}^{2(2^p-1)} m_{3,j} \quad , \end{aligned} \tag{39}$$

where we explicitly allow for individual block sizes $m_{3,j}$. As shown in Fig. 13 b, the decomposition of $m_{3,j+1}$ control qubits (solid boxes) and $m_{3,j}$ spacer qubits (dashed boxes) leads to an auxiliary gate, which we term $\mathrm{C}^{(1+m_{3,j+1})}\mathrm{NOT}^{(m_{3,j})}$. It can be realised as in Fig. 14. Note that the construction scheme of Fig. 10 a requires to each solid box an equally sized dashed box.

In order to express the overall duration, we need the following notation: let the array $\vec{m}_3 := (\vec{m}_3^1, \vec{m}_3^2, \vec{m}_3^3, \vec{m}_3^4)$ of total length $2\ell - 2$ comprise the box sizes $m_{3,j}$ of Fig. 13 b grouped into the four subsets

$\vec{m}_3^1$: sizes of the $\frac{\ell}{2}$ solid boxes on the left,

$\vec{m}_3^2$: sizes of the $\frac{\ell-2}{2}$ solid boxes on the right,

$\vec{m}_3^3$: sizes of the $\frac{\ell}{2}$ dashed boxes on the left, and

$\vec{m}_3^4$: sizes of the $\frac{\ell-2}{2}$ dashed boxes on the right,

and let $\overline{m_3^s}$ be the largest entry in $\vec{m}_3^s$, $s = 1, 2, 3, 4$. Then the duration of the decomposition of a C^{n-2}NOT-gate of Fig. 10 a according to Fig. 13 and 14 reads

$$\begin{aligned} \tau(\mathrm{C}^{n-2}\mathrm{NOT}) = \quad & 2 \cdot 2^p \Big(\tau(\mathrm{C}^{\overline{m_3^1}+1}\mathrm{NOT}) + \tau(\mathrm{C}^{\overline{m_3^2}+1}\mathrm{NOT}) \\ & \qquad + 2\max(\overline{m_3^3}, \overline{m_3^4}) \cdot \tau(\mathrm{CNOT}) \Big) \\ & + 2 \cdot 2^p \Big(\tau(\mathrm{C}^{\overline{m_3^3}+1}\mathrm{NOT}) + \tau(\mathrm{C}^{\overline{m_3^4}+1}\mathrm{NOT}) \\ & \qquad + 2\max(\overline{m_3^1}, \overline{m_3^2}) \cdot \tau(\mathrm{CNOT}) \Big) \\ & + \tau_{\mathrm{SWAP}_{1,m_2}} \ . \end{aligned} \tag{40}$$

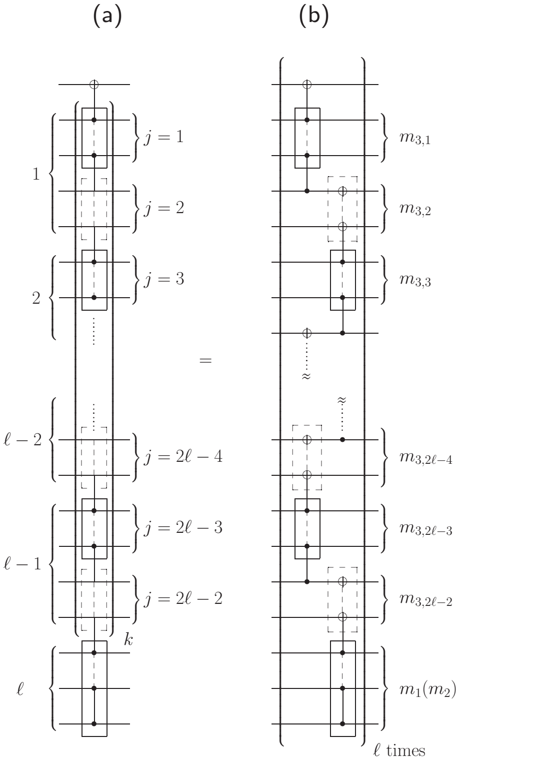

Figure 13: (a) Alternative decomposition of the constituent module of Fig. 10 a. The qubits tagged by the numbers $1, 2, \ldots, \ell$ relate to the neutral qubits of Fig. 12. (b) Each auxiliary building block involves a solid box containing $m_{3,j+1}$ control qubits and a dashed box containing $m_{3,j}$ spacer qubits. It is termed $\mathrm{C}^{(1+m_{3,j+1})}\mathrm{NOT}^{(m_{3,j})}$ and its realisation is shown in Fig. 14.

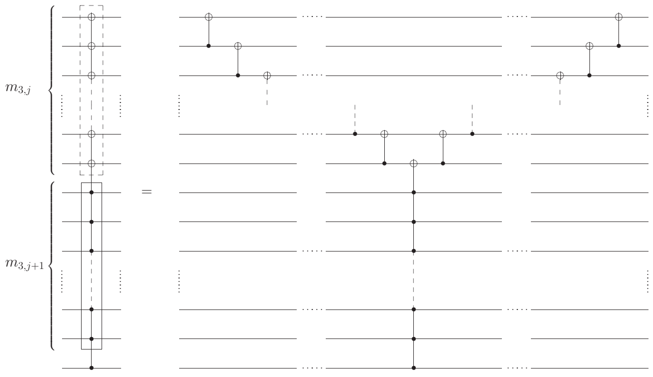

Figure 14: Realisation of the auxiliary building block involving several NOT actions.

Obviously Eqn. 40 is symmetric under the exchanges $\overline{m_3^1} \leftrightarrow \overline{m_3^2}$ and $\overline{m_3^3} \leftrightarrow \overline{m_3^4}$, while the max-functions break a full symmetry that would also require invariance under $\overline{m_3^1} \leftrightarrow \overline{m_3^3}$ and $\overline{m_3^2} \leftrightarrow \overline{m_3^4}$. Consequently, the broken symmetry imposes rules how to increase the box sizes in a time saving way. However, since the duration is limited by the largest box size in each time slot (left part and right part in Fig. 13 b), one can fill the left and the right slots sequentially.

Given $n-2$ control qubits, a time saving decomposition of a $\mathrm{C}^{n-2}\mathrm{NOT}$ results by following the subsequent rules: calculate the auxiliary variables $p' := \lfloor \log_2(n-1) \rfloor - 1$ and $r := (n-1) - 2^{p'+1}$ to determine p as

1. $\mathbf{p = p' - 1}$ for $p' \geq 2$ and $r \in [1, 2^{p'-2}]$ entailing $m_{3,j} \in \{2, 3\}$ and the jump at half width within step I (see Fig. 15);

2. $\mathbf{p = p' - 2}$ for $p' \geq 3$ and $r \in [1+2^{p'-1}, 5 \cdot 2^{p'-3}]$ entailing $m_{3,j} \in \{5, 6\}$ and the minor jump at quarter width within step II (Fig. 15);

3. $\mathbf{p = p'}$ otherwise; then $m_{3,j} \in \{1, 2\}$.

4. NB: for $p = p' - 3$ blocksizes would increase to $m_{3,j} \in \{8, 9\}$ leading to $\mathrm{C}^9\mathrm{NOT}$ and $\mathrm{C}^{10}\mathrm{NOT}$ building blocks, which are currently out of reach.

Then a time-saving decomposition obeys the final rule

5. Once p and the box sizes $m_{3,j} \in \{b, b+1\}$ are fixed, arrange the vector of grouped sizes

$$\begin{aligned}\vec{m}_3 &:= (\vec{m}_3^1, \vec{m}_3^2, \vec{m}_3^3, \vec{m}_3^4) \\ &= (b+1, b+1, \dots, b+1, b, b, \dots, b, b)\end{aligned}$$

with the entries in descending order.

Clearly, the duration will not increase as long as all entries $(b+1)$ fall into $\vec{m}_3^1$, where one may choose to start on top of Fig. 13. A time-step will occur as soon as the first $(b+1)$ falls into $\vec{m}_3^2$, which neither affects $\max(\overline{m_3^1}, \overline{m_3^2})$ nor $\max(\overline{m_3^3}, \overline{m_3^4})$. Analogous features hold for filling $\vec{m}_3^3$ and $\vec{m}_3^4$. They bring about the periodic step function shown in Fig. 11. Its details are given in Fig. 15, where the jump within step I is due to rule 1, while the minor jump within step II has its roots in rule 2 above.

For illustration, consider the following three cases:

Example 1 *In a system of $n = 41$ qubits a $\mathrm{C}^{39}\mathrm{NOT}$ with one auxiliary qubit gives $p' = 4$. By $r = 8$ rule 3 applies and $p = 4$. Eqn. 39 and rule 5 then yield $\vec{m}_3^1 = (2)_8$ [70] and $\vec{m}_3^3 = (1)_8$, while $\vec{m}_3^2 = \vec{m}_3^4 = (1)_7$ and $m_1 = m_2 = 1$. Hence the time-saving decomposition involves just* CNOT *and Toffoli gates (beyond the auxiliary* CNOT *and indirect* SWAP *gates).*

Example 2 *Yet for $n = 42$ qubits a $\mathrm{C}^{40}\mathrm{NOT}$ gives $p' = 4$ and $r = 9 \in [1 + 2^{p'-2}, 5 \cdot 2^{p'-4}] = [9, 10]$, so rule 2 applies and yields $p = 2$. By Eqn. 39 and rule 5 one finds $\vec{m}_3^1 = (6, 5)$ and $\vec{m}_3^3 = (5, 5)$ as well as $\vec{m}_3^2 = (5)$ and $\vec{m}_3^4 = (5)$ with $m_1 = 5$ and $m_2 = 5$. So the $\mathrm{C}^{40}\mathrm{NOT}$ decomposes favourably via $\mathrm{C}^6\mathrm{NOT}$ and $\mathrm{C}^7\mathrm{NOT}$ gates.*

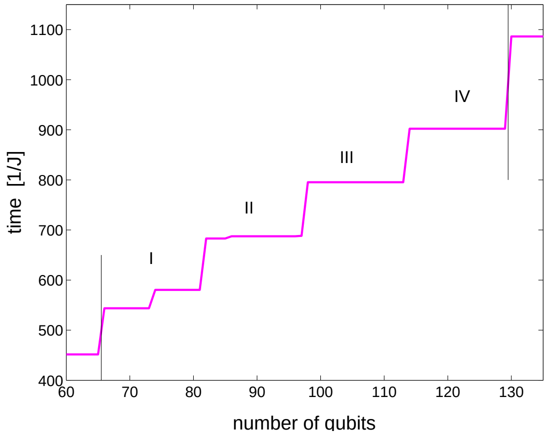

Figure 15: (Colour online) Detailed blow-up of the box in Fig. 11. The step function is a periodic repetition of steps I-IV, where the length of the steps (in units of numbers of qubits) doubles with every period as indicated by the tags in Fig. 11. The small jumps within steps I and II are explained in the text.

Example 3 *Finally, in a system of $n = 137$ qubits a $\mathrm{C}^{135}\mathrm{NOT}$ gives $p' = 6$ and $r = 8 \in [1, 2^{p'-2}] = [1, 16]$, so rule 1 applies and $p = 5$. Eqn. 39 and rule 5 then give $\vec{m}_3^1 = \big((3)_8, (2)_8\big)$ and $\vec{m}_3^3 = (2)_{16}$, $\vec{m}_3^2 = \vec{m}_3^4 = (2)_{15}$, $m_1 = m_2 = 2$. Therefore assembling a $\mathrm{C}^{135}\mathrm{NOT}$ refers to $\mathrm{C}^4\mathrm{NOT}$ and $\mathrm{C}^3\mathrm{NOT}$ gates.*

Finally, the times of Eqn. 40 as well as the decomposition schemes of Fig. 10 a, Fig. 13 and Fig. 14 translate into the respective quality factors as

$$\begin{aligned} q(\mathrm{C}^{n-2}\mathrm{NOT}) &= \Big(F_{\mathrm{C}^{1+m_1}\mathrm{NOT}} \cdot F_{\mathrm{C}^{1+m_2}\mathrm{NOT}} \cdot F_{\mathrm{C}^{1+m_{3,1}}\mathrm{NOT}}\Big)^{2^{p+1}} \\ &\times \prod_{j=2}^{\ell-1} \Big\{ \Big(F_{\mathrm{C}^{1+m_{3,2j-1}}\mathrm{NOT}}\Big)^{2^{p+1}} \cdot \big(F_{\mathrm{CNOT}}\big)^{2^{p+2}\cdot(m_{3,2j-2}-1)} \Big\} \\ &\times \Big(F_{\mathrm{C}^{1+m_{3,2\ell-2}}\mathrm{NOT}}\Big)^{2^{p+1}} \\ &\times \prod_{j=2}^{\ell-1} \Big\{ \Big(F_{\mathrm{C}^{1+m_{3,2\ell+2-2j}}\mathrm{NOT}}\Big)^{2^{p+1}} \cdot \big(F_{\mathrm{CNOT}}\big)^{2^{p+2}\cdot(m_{3,2\ell+1-2j}-1)} \Big\} \\ &\times \Big(F_{\mathrm{SWAP}_{1,m_1+1}}\Big)^4 \times \Big(F_{\mathrm{SWAP}_{1,m_2}}\Big)^2 \times e^{-\tau(\mathrm{C}^{n-2}\mathrm{NOT})/T_R} \;. \end{aligned} \tag{41}$$

The corresponding numerical quality results have already been shown in Fig. 11 b as step function. They represent compilations with building blocks using multiply controlled subblocks with up to 6 and 7 control qubits thus giving another significant improvement over the assembly scheme described in the previous subsection. The results are also summarised in Tab. II.

IV. IMPLICATIONS FOR MULTIPLY-CONTROLLED GENERAL UNITARY OPERATIONS

The fast assembly schemes of multiply controlled NOT gates given in the previous subsection also allow for faster realisations of multiply controlled general unitary gates than in the classic of Barenco, Bennett *et al.*, [20].

Lemma 1 *Recall: every self-inverse 1-qubit special unitary $U_+ = U_+^{-1} \in SU(2)$ is trivially $\pm\mathbb{1}$, while every self-inverse $U_- \in U(2) \setminus SU(2)$ is unitarily similar to σ_x.*

Proof: To see the second assertion constructively, observe that any self-adjoint $U_- \in U(2) \setminus SU(2)$ shows $\det U_- = -1$ and may thus be represented as a pure quaternion $U_- = x{\cdot}\sigma_x + y{\cdot}\sigma_y + z{\cdot}\sigma_z$ with $x^2+y^2+z^2 = 1$. Ensuring $|a|^2 + |b|^2 = 1$ in $V \in SU(N)$, it can be identified with

$$U_- = V\sigma_x V^\dagger = \begin{pmatrix} a & b \\ -b^* & a^* \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} a^* & -b \\ b^* & a \end{pmatrix} \qquad (42)$$

via $x = \mathrm{Re}(a^2 - b^2)$, $y = \mathrm{Im}(b^2 - a^2)$, $z = 2\,\mathrm{Re}(ab^*)$. ■

Corollary 2 *Given a realisation of a* $\mathrm{C}^{n-2}\mathrm{NOT}$ *in time* $\tau(\mathrm{C}^{n-2}\mathrm{NOT})$ *on an n-qubit system with one auxiliary and one target qubit. Then the realisation of a multiply controlled general unitary* $\mathrm{C}^{n-2}\mathrm{U}$ *takes the time*

1. $\tau(\mathrm{C}^{n-2}\mathrm{U}) \leq \tau(\mathrm{C}^{n-2}\mathrm{NOT}) + \tau(V) + \tau(V^\dagger)$,

   if $U \in U(2) \setminus SU(2)$ is self-inverse $U^2 = \mathbb{1}_2$ and $V \in SU(2)$ as in Eqn. 42;

2. $\tau(\mathrm{C}^{n-2}\mathrm{U}) \leq 2 \cdot \tau(\mathrm{C}^{n-2}\mathrm{NOT}) + \tau(3 \;\; local \;\; gates)$,

   if $U \in SU(2)$ and $U^2 \neq \mathbb{1}$;

3. *Assertion (1) can be generalised to multiply-controlled $(q+1)$-qubit self-inverse unitaries of the form $U_- = V(\sigma_x \otimes \mathbb{1}_{2^q})V^\dagger$ with $V \in SU(2^{q+1})$.*

Proof:

(1) The inequality is a direct consequence of applying Eqn. 42 to the NOT operation on the target qubit.This is qubit 1 in Fig. 10 a, which for later convenience becomes qubit 0 in Figs. 12 and 13. (Moreover, by reversing Eqn. 42 to $\sigma_x = V^\dagger U_- V$ and using it on qubit 0 in Fig. 12, one can absorb the time for at least one of the local gates V by virtue of the decomposition of Fig. 13 b.)

(2) Direct consequence of Lemma 5.1 in Ref. [20].

(3) Obvious generalisation of Eqn. 42 with q further qubits added—e.g., on top of qubit 1 in Fig. 10 a. ■

Similar generalisations hold for further special cases addressed in Ref. [20] Sec. 5.2.

V. CONCLUSIONS AND OUTLOOK

We have exploited the power of a cutting-edge high-performance parallel cluster for quantum CISC compilation. Thus the standard toolbox of universal quantum gate modules (RISC) has been extended by time-optimised effective multi-qubit gates (CISC). We have shown ways how these CISC modules can be assembled in a scalable way in order to address large-scale quantum computing on systems that are too large for direct CISC compilation. Although our optimal-control based CISC-compilation routine exploits parallel matrix operations for clusters as well as fast matrix exponentials [58], increasing the system size by one qubit roughly requires a factor of eight more CPU time. Since direct CISC compilation thus scales exponentially, scalable assembler schemes for multi-qubit CISC modules are needed, and we have presented some elementary yet important ones:

For indirect SWAPs, the quantum Fourier transform, and multiply-controlled NOT -gates, the CISC decomposition is significantly faster than the standard RISC decomposition into local plus universal two-qubit gates. The current improvements range from 20% up to a speed-up by nearly 300%. However, as illustrated in Tab. II, the potential of CISC compilation is by far not yet exhausted with the current schemes. — As a noteworthy side result, we have shown that gate errors in multi-qubit CISC modules propagate more favourably than in RISC modules confined to two-qubit gates.

Assembling pre-compiled effective multi-qubit modules has further advantages beyond saving gate time: a problem common to many implementations occurs as soon as smaller decoherence-protected modules shall be embedded in larger effective systems. Usually dissipative coupling to a new environment also introduces new sources of decay the original module has not been optimised for. Therefore, practical handling is greatly facilitated, if the m-qubit modules with tailored optimisation under dissipation and decoherence (see, e.g., [43]) extend to larger units of relaxatively interacting qubits than the standard of $m = 2$. This advantage can readily be envisaged by a quantum-information processor, e.g., based on trapped ions, where the 'passive qubits' are stored with spatial separation from the currently 'active ones' in the processing unit (see, e.g., [67, 68, 69] for overview and recent developments).

Moreover, controlling physical m-qubit modules will also allow for encoding logical qubits with specifically tailored optimisiation under more realistic relaxation models [43] than in ideal 'decoherence-free subspaces'.

This paves the way to another frontier of research: optimising the quantum assembler task on the extended toolbox of quantum CISC-modules with effective many-qubit interactions. Finally, it is to be anticipated that methods developed in classical computer science can also be put to good use for systematically optimising quantum assemblers.

Table II: Current Exploitation vs. Potential of Quantum CISC Compilation (Limit of Fast Local Controls)

gate	CISC potential:	estimate	current status:	exploitation	improvement
		$\pi_{\text{CISC}} = \Delta_2/\Delta_\infty$		$\eta_m = \Delta_\infty/\Delta_m$	$\xi_m = \Delta_2/\Delta_m$
$\text{SWAP}_{1,n}$	medium	2.16	fairly exhausted	0.86 $[m = 8]$	1.88
n-QFT	medium	2.27	halfway exhausted	0.53 $[\mathbf{m} = (5_{\text{QFT}}, 10_{\text{CP-SWAP}})]$	1.20
C^{n-2}NOT	big	13.6	not yet exhausted	0.18 $[\mathbf{m} = (8,8,1)_{n \text{ odd}}$ or $(8,7,1)_{n \text{ even}}]$	2.45
				0.25–0.31 $[\mathbf{m} = (\leq 7, \leq 6, \leq 6)]$	3.45–4.19

Acknowledgments

This work builds upon parallel programming reported in Refs. [47, 58]: we wish to thank K. Waldherr, T. Gradl, and T. Huckle for implementing further inprovements beyond the above references. The work has been supported in part by the integrated EU project QAP, by the Bavarian programme of excellence QCCC, and by *Deutsche Forschungsgemeinschaft*, DFG, within the incentive SFB-631. Access to the high-performance parallel cluster HLRB-II at *Leibniz Rechenzentrum* of the Bavarian Academy of Science is gratefully acknowledged. We wish to thank Matthias Christandl and Robert Zeier for fruitful discussions.

[1] R. P. Feynman, Int. J. Theo. Phys. **21**, 467 (1982).

[2] R. P. Feynman, *Feynman Lectures on Computation* (Perseus Books, Reading, MA., 1996).

[3] P. W. Shor, in *Proceedings of the Symposium on the Foundations of Computer Science, 1994, Los Alamitos, California* (IEEE Computer Society Press, New York, 1994), pp. 124–134.

[4] P. W. Shor, SIAM J. Comput. **26**, 1484 (1997).

[5] C. H. Papadimitriou, *Computational Complexity* (Addison Wesley, Reading, MA., 1995).

[6] R. Jozsa, Proc. R. Soc. A. **454**, 323 (1998).

[7] R. Cleve, A. Ekert, C. Macchiavello, and M. Mosca, Proc. R. Soc. A. **454**, 339 (1998).

[8] M. Ettinger, P. Høyer, and E. Knill, Inf. Process. Lett. **91**, 43 (2004).

[9] J. W. Cooley and J. W. Tukey, Math. Comput. **19**, 297 (1965).

[10] T. Beth, *Verfahren der schnellen Fourier-Transformation* (Teubner, Stuttgart, 1984).

[11] S. Lloyd, Science **273**, 1073 (1996).

[12] D. Abrams and S. Lloyd, Phys. Rev. Lett. **79**, 2586 (1997).

[13] C. Zalka, Proc. R. Soc. London A **454**, 313 (1998).

[14] C. Bennett, I. Cirac, M. Leifer, D. Leung, N. Linden, S. Popescu, and G. Vidal, Phys. Rev. A **66**, 012305 (2002).

[15] L. Masanes, G. Vidal, and J. Latorre, Quant. Inf. Comput. **2**, 285 (2002).

[16] E. Jané, G. Vidal, W. Dür, P. Zoller, and J. Cirac, Quant. Inf. Computation **3**, 15 (2003).

[17] D. Deutsch, Proc. Royal Soc. London A **400**, 97 (1985).

[18] J. Dodd, M. Nielsen, M. Bremner, and R. Thew, Phys. Rev. A **65**, 040301(R) (2002).

[19] M. Bremner, D. Bacon, and M. Nielsen, Phys. Rev. A **71**, 052312 (2005).

[20] A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. W. Shor, T. Sleator, J. A. Smolin, and H. Weinfurter, Phys. Rev. A **52**, 3457 (1995).

[21] E. Knill (1995), LANL report LAUR-95-2225 available as e-print: http://arXiv.org/pdf/quant-ph/9508006.

[22] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge (UK), 2000).

[23] A. Y. Kitaev, A. H. Shen, and M. N. Vyalyi, *Classical and Quantum Computation* (American Mathematical Society, Providence, 2002).

[24] N. D. Mermin, *Quantum Computer Science: An Introduction* (Cambridge University Press, Cambridge, 2007).

[25] A. Blais, Phys. Rev. A **64**, 022312 (2001).

[26] R. Solovay (1995), unpublished personal communication to A. Yu. Kitaev.

[27] A. Y. Kitaev, Russ. Math. Surveys **52**, 1191 (1997), Russian original: *Uspekhi Mat. Nauk.* **52**, 53–112.

[28] M. A. Nielsen, M. R. Dowling, M. Gu, and A. C. Doherty, Science **311**, 1133 (2006).

[29] M. A. Nielsen, Quant. Inf. Computation **6**, 213 (2006).

[30] G. Vidal, K. Hammerer, and J. I. Cirac, Phys. Rev. Lett. **88**, 237902 (2002).

[31] A. M. Childs, H. L. Haselgrove, and M. A. Nielsen, Phys. Rev. A **68**, 052311 (2003).

[32] R. Zeier, M. Grassl, and T. Beth, Phys. Rev. A **70**, 032319 (2004).

[33] P. Wocjan, D. Janzing, and T. Beth, Quant. Inf. Comput. **2**, 117 (2002).

[34] T. Schulte-Herbrüggen, A. K. Spörl, N. Khaneja, and S. J. Glaser, Phys. Rev. A **72**, 042331 (2005).

[35] V. Ramakrishna and H. Rabitz, Phys. Rev. A **54**, 1715 (1995).

[36] T. Schulte-Herbrüggen, *Aspects and Prospects of High-Resolution NMR* (PhD Thesis, Diss-ETH 12752, Zürich, 1998).

[37] S. J. Glaser, T. Schulte-Herbrüggen, M. Sieveking, O. Schedletzky, N. C. Nielsen, O. W. Sørensen, and C. Griesinger, Science **280**, 421 (1998).

[38] R. R. Tucci (1999), e-print: http://arXiv.org/pdf/quant-ph/9902062.

[39] D. Williams, *Quantum Computer Architecture, Assembly Language and Compilation* (Master's Thesis, University of Warwick, 2004).

[40] V. V. Shende, S. Bullock, and I. L. Markov, IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. **25**, 1000 (2006).

[41] K. M. Svore, A. V. Aho, A. W. Cross, I. Chuang, and I. L. Markov, Computer **25**, 74 (2006).

[42] R. R. Tucci (2007), e-print: http://arXiv.org/0706.0479.

[43] T. Schulte-Herbrüggen, A. Spörl, N. Khaneja, and S. Glaser (2006), e-print: http://arXiv.org/pdf/quant-ph/0609037.

[44] G. D. Sanders, K. W. Kim, and W. C. Holton, Phys. Rev. A **59**, 1098 (1999).

[45] N. Khaneja, T. Reiss, C. Kehlet, T. Schulte-Herbrüggen, and S. J. Glaser, J. Magn. Reson. **172**, 296 (2005).

[46] A. K. Spörl, T. Schulte-Herbrüggen, S. J. Glaser, V. Bergholm, M. J. Storcz, J. Ferber, and F. K. Wilhelm, Phys. Rev. A **75**, 012302 (2007).

[47] T. Gradl, A. K. Spörl, T. Huckle, S. J. Glaser, and T. Schulte-Herbrüggen, Lect. Notes Comput. Sci. **4128**, 751 (2006), Proceedings of the EURO-PAR 2006.

[48] V. Jurdjevic and H. Sussmann, J. Diff. Equat. **12**, 313 (1972).

[49] T. Schulte-Herbrüggen, K. Hüper, U. Helmke, and S. J. Glaser, *Applications of Geometric Algebra in Computer Science and Engineering* (Birkhäuser, Boston, 2002), chap. Geometry of Quantum Computing by Hamiltonian Dynamics of Spin Ensembles, pp. 271–283.

[50] F. Albertini and D. D'Alessandro, Lin. Alg. Appl. **350**, 213 (2002).

[51] F. Albertini and D. D'Alessandro, IEEE Trans. Automat. Control **48**, 1399 (2003).

[52] D. Gross, S. Flammia, and J. Eisert (2008), e-print: http://arXiv.org/pdf/0810.4331.

[53] C. Dankert, R. Cleve, J. Emerson, and E. Livine (2006), http://arXiv.org/quant-ph/0606161.

[54] D. Gross, K. Audenaert, and J. Eisert, J. Math. Phys. **48**, 052104 (2007).

[55] A. Ambainis and J. Emerson, Proc. Complexity 2007 pp. 129–140 (2007), see also http://arXiv.org/quant-ph/0701126.

[56] T. Schulte-Herbrüggen, S. J. Glaser, G. Dirr, and U. Helmke (2008), e-print: http://arXiv.org/pdf/0802.4195.

[57] F. Mezzadri, Notices Amer. Math. Soc. **54**, 592 (2007).

[58] T. Schulte-Herbrüggen, A. K. Spörl, K. Waldherr, T. Gradl, S. J. Glaser, and T. Huckle, *in: High-Performance Computing in Science and Engineering, Garching 2007* (Springer, Berlin, 2008), chap. Using the HLRB Cluster as Quantum CISC Compiler: Matrix Methods and Applications for Advanced Quantum Control by Gradient-Flow Algorithms on Parallel Clusters, pp. 517–533.

[59] K. Waldherr, *Die Matrix-Exponentialabbildung: Eigenschaften und Algorithmen* (Diploma Thesis, Technical University Munich, 2007).

[60] N. Khaneja, R. Brockett, and S. J. Glaser, Phys. Rev. A **63**, 032308 (2001).

[61] N. Khaneja, S. J. Glaser, and R. Brockett, Phys. Rev. A **65**, 032301 (2002).

[62] A. Saito, K. Kioi, Y. Akagi, N. Hashizume, and K. Ohta (2000), quant-ph/0001113.

[63] R. Zeier (2007), personal communication.

[64] M. Clausen and U. Baum, *Fast Fourier Transforms* (Bibliographisches Institut, Mannheim, 1993).

[65] S. Egner, *Zur algorithmischen Zerlegungstheorie linearer Transformationen mit Symmetrie* (PhD Thesis, University of Karlsruhe, 1997).

[66] E. Lucas, Am. J. Math. **1**, 184 (1878), see also: M. Sved, *Math. Intelligencer* **10** (1988), 56–64 and R. Bollinger and C. Burchard, *Am. Math. Monthly* **97** (1990), 198–204.

[67] R. Blatt and D. Wineland, Nature (London) **453**, 1008 (2008).

[68] J. Chiaverini, M. D. Barrett, R. B. Blakestad, J. Britton, J. D. Jost, E. Knill, C. Langer, D. Leibfried, R. Ozeri, T. Schätz, et al., Proc. SPIE **6256**, 625610 (2006).

[69] R. Maiwald, G. Leuchs, D. Leibfried, J. Britton, J. C. Bergquist, and D. Wineland (2008), http://arXiv.org/pdf/0810.2647.

[70] The indices shall serve as a short-hand to denote, e.g., $(5)_2 := (5,5) \in \mathbb{Z}_2$.

APPENDIX SECTION

A. Variant-II of Scalably Assembling a Quantum Fourier Transform

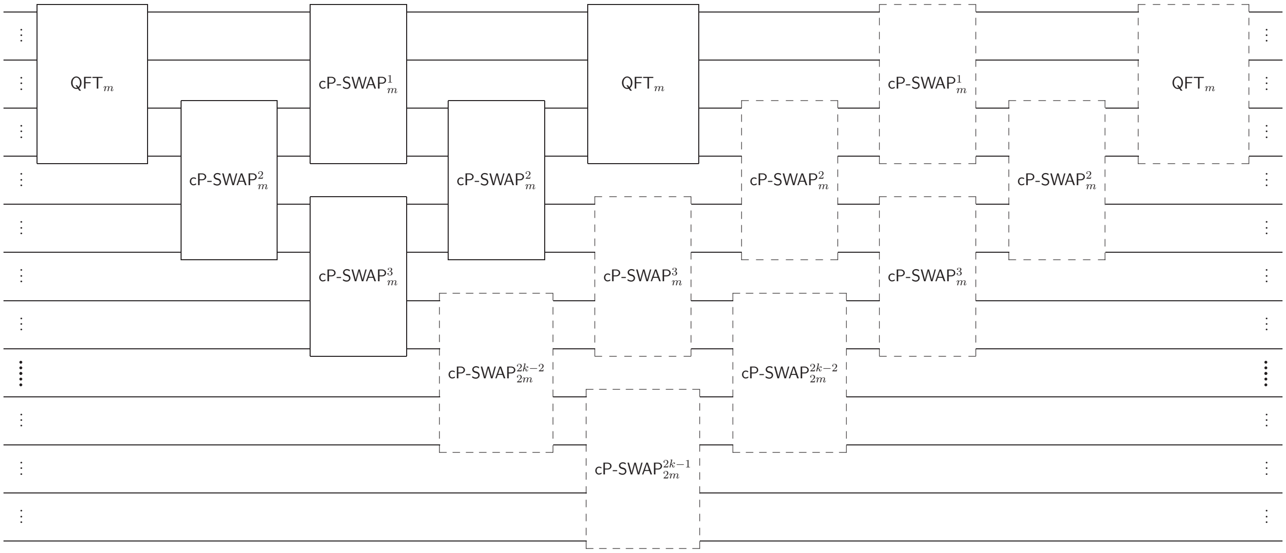

Figure 16: For $k \geq 2$ and even m, a (km)-qubit QFT can be assembled from k times an m-qubit QFT and $4\binom{k}{2}$ instances of m-qubit modules cP-SWAP^j_m, where the index j of different phase-rotation angles takes the values $j = 1, 2, \ldots, 2k-1$. The dashed boxes correspond to Fig. 7 and show the induction $k \mapsto k+1$.

B. Controlled-Phase-SWAP Modules for $k \cdot 10$-Qubit QFTs

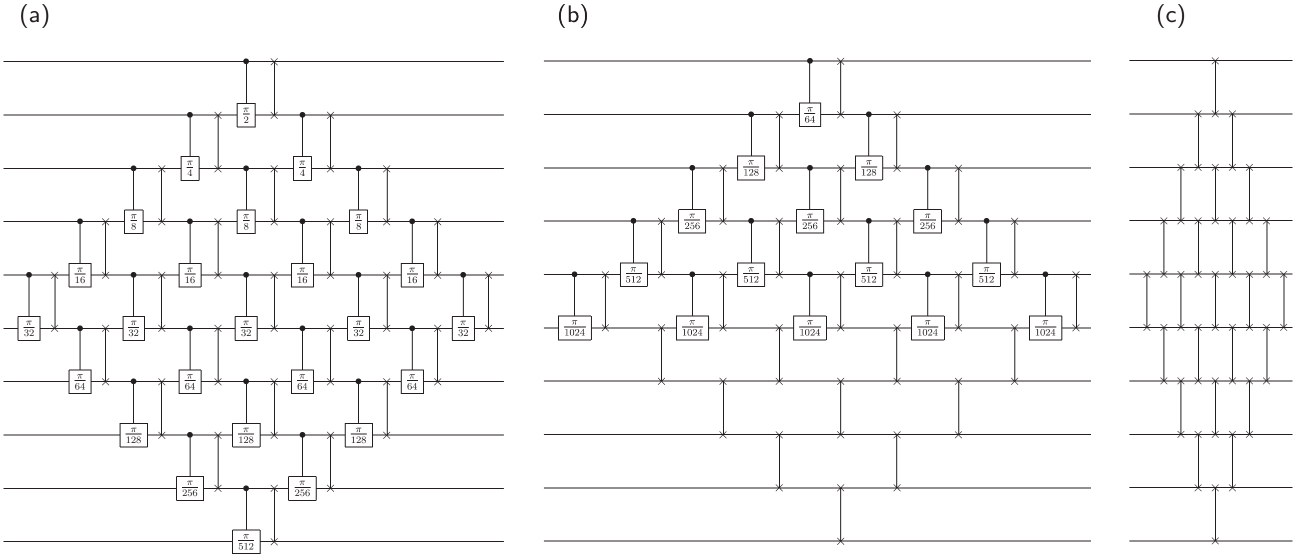

Figure 17: Equivalent circuit representations of the three 10-qubit cP-SWAP modules needed for a $k \cdot 10$-qubit QFT, when rotation angles less than $\pi/2^{10}$ are omitted (as described in the text): (a) cP-SWAP$^1_{10}$ with no truncation so $F_{tr} = 1$, (b) cP-SWAP$^2_{10}$ with $F_{tr} = 0.9999902$, and (c) cP-SWAP$^j_{10}$, which covers all $j \geq 3$ with fidelity $F_{tr} \geq 0.9999991$. These building blocks are compiled directly as effective 10-qubit modules thus reducing the time to less than 60% of the duration required for the decomposition into 2-qubit modules.